

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matevž Dolinar

**Model vrednotenja procesa razvoja
programske opreme in njegova
uporaba v podjetju**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Tomaž Hovelja

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kandidat naj preuči literaturo o vrednotenju procesa razvoja programske opreme. Na osnovi tako pridobljenega znanja naj oblikuje model vrednotenja procesa razvoja programske opreme. Kandidat naj oblikovani model preizkusi v podjetju. Poudarek dela v diplomski nalogi naj bo na praktični uporabnosti oblikovanega modela.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matevž Dolinar, z vpisno številko **63080058**, sem avtor diplomskega dela z naslovom:

Model vrednotenja procesa razvoja programske opreme in njegova uporaba v podjetju

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Tomaža Hovelje
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 17. septembra 2014

Podpis avtorja:

Neizmerno se zahvaljujem mentorju doc. dr. Tomažu Hovelji za pomoč, usmerjanje in svetovanje pri izdelavi diplomskega dela. Zahvale gredo tudi sodelavcem, ki so mi omogočili izvedbo študije primera. Zahvalil bi se tudi družini in prijateljem, ki so mi vedno stali ob strani. Posebna zahvala gre Manji, ki je poskrbela za manjkajoče vejice in me spodbujala ter motivirala pri izdelavi tega dela. Hvala vsem.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Pregled literature	5
3	Model vrednotenja razvojnega procesa	11
3.1	Gradnja modela	11
3.2	Ocenjevanje aktivnosti	13
3.3	Ogrodje modela	15
4	Študija primera	19
4.1	Storitev A	20
4.2	Storitev B	26
4.3	Storitev C	31
4.4	Storitev D	37
5	Sklepne ugotovitve	43
A	Popis aktivnosti	49
A.1	Storitev A	49
A.2	Storitev B	52
A.3	Storitev C	56

KAZALO

A.4 Storitve D	58
B Primer vprašalnika za management	61
C Primer vprašalnika za tehnične vodje	65
D Primer vprašalnika za uporabnike SDM	67

Seznam uporabljenih kratic

kratica	angleško	slovensko
SDM	software developement methodology	metodologija razvoja programske opreme
DOI	diffusion of innovation	teorija o difuziji inovacij
TAM	technology acceptance model	model sprejetja tehnologije
PCI	perceived characteristics of innovating	zaznane lastnosti inovacij
TRA	theory of reasoned action	teorija utemeljenega ukrepanja
IT	information technology	informacijska tehnologija

Povzetek

Skozi bogato zgodovino področja ocenjevanja sprejetja metodologije je bilo predlaganih več različnih teorij in modelov vrednotenja. Obstoječe rešitve v veliki meri zanemarljajo poslovni aspekt metodologije, zaradi česar je vloga managementa pri odločitvah povezanih s SDM (ang. software development methodology) majhna, posledično pa podjetja ne izkoriščajo potencialnih koristi uporabe metodologij razvoja programske opreme. V diplomskem delu bo zato predlagan model vrednotenja razvojnega procesa, ki združuje socialno, tehnično in ekonomsko dimenzijo. Predlagan model bo razvojni proces ocenjeval na nivoju aktivnosti, kar ni stalna praksa obstoječih modelov, ki metodologijo večinoma obravnavajo kot celoto. Model s tem omogoča enostavno prepoznavanje kritičnih aktivnosti, kar managerjem omogoči lažje odločanje pri naložbah v SDM. Model je bil v praksi preizkušen na primeru slovenskega podjetja. Rezultati so potrdili tako smiselnost ocenjevanja aktivnosti kot tudi uporabo treh različnih dimenzij vrednotenja.

Ključne besede: model vrednotenja SDM, socialna sprejetost SDM, tehnična primernost SDM, vpliv SDM na uspešnost podjetja

Abstract

To date, many software development methodology evaluation models and theories have been proposed. Most of them, however, neglect the economical aspect of SDM, which results in management having little role in decisions regarding SDM. Consequently enterprises do not reap all the potential benefits that SDM adoption may bring. In this thesis a SDM evaluation model that combines social, technical and economical aspect of SDM is introduced. Proposed model makes evaluation based on SDM activities, unlike existing models that focus specifically on SDM as a whole. This facilitates identifying critical activities and helps managers at making better decisions regarding SDM investments. The model was tested on a practical case of Slovenian company. The results confirmed that evaluating individual SDM activities as well as making the evaluation using the three aspects is a reasonable approach.

Keywords: SDM evaluation model, SDM adoption, technical efficiency of SDM, economical impact of SDM

Poglavje 1

Uvod

Metodologija razvoja programske opreme (ang. *Software Development Methodology* - SDM) predstavlja teoretično ogrodje, ki se uporablja pri planiranju, izvajanju in nadziranju razvoja programske opreme. Definirana je kot skupek postopkov, aktivnosti, nalog, pravil, dokumentacije in orodij, ki se jih uporabi v procesu razvoja [1]. Glede na to lahko definiramo sprejetost SDM kot stopnjo konsistentnosti, s katero se metode, ki jih predlaga metodologija, uporabljajo v danem podjetju ali projektu razvoja programske opreme [14]. Večina podjetij, ki se ukvarja z razvijanjem programske opreme med razvojem sledi določenim smernicam in pravilom [14]. Le-ta so lahko jasno določena in zapisana - v tem primeru govorimo o uporabi formalizirane SDM. A v praksi se precej bolj pogosto uporablja druga preprostejša možnost – razvijalci se pri svojem delu zanašajo zgolj na neformalne dogovore. Slednje je v nasprotju s študijami, ki razkrivajo, da uporaba formalizirane metodologije poviša tako produktivnost kot tudi kvaliteto razvoja [1, 3]. Ena glavnih koristi, ki jih metodologija prinaša je razbitje kompleksnega razvojnega procesa, kar olajša projektno načrtovanje in nadzor nad razvojem [4]. S standardizacijo razvojnega procesa omogoča tudi hranjenje in sistemizacijo izkušenj iz preteklih procesov razvoja za prihodnjo rabo. Slednje pripomore k lažji zamenljivosti razvijalcev v procesu razvoja. Uporaba metodologije prav tako poviša produktivnost in kakovost razvoja, saj z njeno uporabo lahko predvi-

dimo potrebe in omogočimo resurse v danem trenutku [2]. Nadvse zanimivo je torej, da kljub temu da formalizirane SDM kot take obstajajo že desetletja, še vedno večina podjetij, ki razvija programsko opremo, odstopa od njihove uporabe [5].

Za slabo sprejetje SDM je bilo ugotovljenih več razlogov [6, 7], med katerimi izstopata dva pomembnejša. Kot prvo, formalizirane SDM niso prirejene za specifične potrebe organizacije ali projekta razvoja. Tak primer je metodologija, ki predpisuje neprimerne aktivnosti in metode: je preveč stroga in se je ne da prilagoditi potrebam projekta razvoja [8, 14]. Drugi razlog za slabo sprejetje je ta, da formalizirana SDM ne ustreza socialnim karakteristikam razvojne ekipe in/ali podjetja. Tak primer je strogo določena metodologija v sicer liberalno naravnem podjetju: inovativna razvojna ekipa, ki odstopa od tradicionalnih prijemov, bo zavrnila uporabo take metodologije [9, 14]. Tehnična in socialna neprimernost sta tako ena od glavnih razlogov zavračanja uporabe formalizirane SDM, saj tako metodologijo razvojna ekipa vidi kot neuporabno in je kljub investiciji podjetja ne bo uporabljala.

Opravljenih je bilo več poskusov izboljšanja sprejetja SDM [10, 11, 14], ki pa so se večinoma končali z jalovim uspehom. Večina teh primerov namreč sprejetje SDM obravnava kot informacijsko odločitev, ki jo morajo sprejeti tehniki in se ne ozira na vlogo managementa podjetja. Največkrat je takega mnenja tudi podjetje samo [12]. Z drugega vidika pa uvedba SDM tipično zavzame zajetno časovno in denarno naložbo, kar govori v prid temu, da bi tudi management moral bolj aktivno sodelovati pri sprejemanju odločitev glede SDM. Večja vključenost managementa bi botrovala sprejemanju boljših odločitev glede naložb v SDM, posledično pa bi podjetja izkoriščala več potencialnih koristi, ki jih uporaba SDM prinaša [1]. Za večjo vključenost managementa v odločitve glede SDM je pomembno boljše razumevanje posledic in vplivov, ki jih SDM prinaša na poslovanje podjetja. Zgolj razumevanje vplivov SDM na uspešnost podjetja managerjem pa ne daje znanja, ki bi vplivalo na sprejemanje pravih odločitev v zvezi s SDM. Potrebno je razumeti tudi stopnjo sprejetja posamezne aktivnosti znotraj metodologije, saj zgolj

zadostno sprejete aktivnosti pozitivno vplivajo na dejanski poslovni uspeh podjetja [13]. Razumevanje obeh vidikov – sprejetja posameznih aktivnosti in pa njihovega vpliva na uspeh podjetja – tako predstavlja osnovo, ki naj bi jo managerji uporabljali za odločanje glede naložb v SDM v podjetju.

Uporabo dotične osnove pa omejujejo že omenjeni razlogi: poslovni aspekt metodologije je v obstoječih poskusih izboljšanja sprejetja SDM največkrat prezrt, zaradi česar je vloga managementa pri odločitvah glede SDM majhna. Hkrati večina obstoječih modelov za ocenjevanje obravnava metodologijo kot celoto, kar managementu ne omogoča razumevanja stopnje sprejetosti in vpliva posameznih aktivnosti metodologije.

V diplomskem delu se bomo reševanja zgoraj predstavljene problematike lotili z razvojem modela vrednotenja razvoja programske opreme, ki bo ocenil sprejetost posamezne aktivnosti metodologije, njeno tehnično primernost in njen vpliv na uspešnost podjetja. Ocenjevanje metodologije na nivoju aktivnosti bo omogočalo lažjo identifikacijo in izboljšanje tistih aktivnosti razvoja programske opreme, ki lahko podjetju prinesejo največ koristi. Z grafičnim prikazom modela bomo managementu omogočili lažje vrednotenje SDM. Predlagali bomo tudi splošne ukrepe, ki naj jih management izvede v namen izboljšanja sprejetosti. Model bo v praksi preizkušen na štirih primerih, z raziskavo pa bomo preverili njegovo uspešnost pri identifikaciji (ne)primernih aktivnosti in snovanju ukrepov, ki managementu omogočajo izboljšanje razvojnega procesa.

Poleg uvoda in zaključka diplomsko delo sestavljajo tri večja poglavja. Najprej je s pregledom literature predstavljeno teoretično ozadje, ki se ga v tretjem poglavju nato uporabi za razvoj modela vrednotenja SDM. Model je nato predstavljen v praksi, kar je prikazano s študijo štirih primerov. Diplomsko delo se zaključuje s sklepnimi ugotovitvami, ki povzamejo ugotovitve raziskave.

Poglavje 2

Pregled literature

V tem poglavju bomo predelali literaturo z namenom identificiranja pomanjkljivosti obstoječih načinov ocenjevanja sprejetosti in vpliva metodologije. Pomembnejša dognanja bomo nato uporabili pri gradnji modela v naslednjem poglavju.

Za raziskave o sprejetju metodologije in ostalih inovacij na področju razvoja programske opreme je zelo pomembna Rogersova teorija o difuziji inovacij (Diffusion of innovation – DOI) [15]. DOI je univerzalna teorija, ki skuša pojasniti, zakaj se določene inovacije med potencialnimi uporabniki razširjajo, spet druge pa ostajajo neuporabljene. Ker določeno metodologijo lahko vidimo tudi kot inovacijo, se raziskave na osnovi prej omenjene teorije osredotočajo na napovedovanje vedenja ciljnih uporabnikov v povezavi z inovacijami [16]. Poleg DOI pa seveda obstajajo še druge teorije za napovedovanje sprejetja inovacij. To so teorija načrtovanega vedenja (TPB – theory of planned behaviour), model sprejetja tehnologij (TAM – technology acceptance model), zaznane lastnosti inovacij (PCI – perceived characteristics of innovating) in teorija utemeljenega ukrepanja (TRA – theory of reasoned action) [13]. Vse omenjene teorije pa se večinoma osredotočajo na sprejetje metodologije iz socioloških, psiholoških ter kulturnih vidikov in tipično zanemarjajo poslovne aspekte sprejetja, ki pa so velikokrat ključnega pomena za management podjetja [13].

Obstoječe raziskave se navadno osredotočajo na opazovanje metodologije kot celote ali pa se podrobno osredotočijo na posamezno aktivnost znotraj metodologije. Tak postopek sicer ni napačen, je pa iz vidika rezultatov precej bolj skop, saj ne omogoča prepoznavanja (ne)ustreznih delov znotraj SDM. Delovno okolje pri razvoju je dinamično in zahteve se stalno spreminjajo, zaradi česar je nemogoče pričakovati sprejetje metodologije s strogo določenimi aktivnostmi. Tudi to govori v prid temu, da je pomembno, da SDM opazujemo kot sestavo prepletajočih se aktivnosti. Dotična filozofija je dodobra podkrepljena tudi z raziskavami na področju situacijskega konstruiranja metodologij [17, 18]. Situacijsko konstruiranje metodologij skuša zgraditi metodologijo, prilagojeno za določeno situacijo, iz aktivnosti že obstoječih SDM. Zaradi tega je pomembno, da se uspešnost in sprejetje SDM lahko oceni na ravni individualnih delov (aktivnosti).

Vavpotič in Hovelja [13] sta stopnjo sprejetja metodologije definirala z meritvijo dveh dimenzij - socialne in tehnične. Socialna dimenzija je predstavljena s frekvenco uporabe aktivnosti SDM v primeru priložnosti za uporabo (FrqUse), za ugotavljanje tehnične primernosti pa se uporabi frekvenca priložnosti za uporabo aktivnosti (FrqOpp). FrqUse izmeri, kako pogosto se razvijalci programske opreme odločijo za uporabo določene aktivnosti SDM v primeru, da se pojavi priložnost za njeno uporabo. FrqOpp pa na drugi strani izmeri, kako pogosto med razvojem prihaja do priložnosti za uporabo določene aktivnosti SDM, ne glede na to, ali se ta dejansko izvede. Dwiwedi je zapisal, da frekvenca uporabe ni primeren indikator za merjenje sprejetosti, kadar je uporaba aktivnosti SDM obvezna [19]. V tem primeru je bolj smiselno kot mero sprejetosti uporabiti zadovoljstvo uporabnikov SDM (t. j. zaposlenih, ki neposredno uporabljajo aktivnosti SDM). Za merjenje zadovoljstva uporabnikov SDM se pogosto uporablja merilne dimenzije, ki sta jih uveljavila Ives in Doll [19]. Te se nanašajo tudi na sistemsko, informacijsko in storitveno kvaliteto, zato so bile za boljše ocenjevanje razvite še dodatne mere, ki se osredotočijo izključno na merjenje zadovoljstva uporabnika. Dimenzije zadovoljstva uporabnikov SDM so prikazane v tabeli 2.1.

Mera	Vir
Ustreznost	Almutairi in Subramanian (2005), Seddon in Yip (1992), Seddon in Kiew (1994)
Učinkovitost	Almutairi in Subramanian (2005), Seddon in Yip (1992), Seddon in Kiew (1994)
Kakovost	Almutairi in Subramanian (2005), Seddon in Yip (1992), Seddon in Kiew (1994)
Užitek pri delu	Gable et al. (2008)
Sistemska kakovost	Gable et al. (2008)
Kakovost informacij	Gable et al. (2008)
Vseprošno zadovoljstvo	Almutairi in Subramanian (2005), Gable et al. (2008), Rai et al. (2002), Seddon in Yip (1992), Seddon in Kiew (1994)

Tabela 2.1: Dimenzije zadovoljstva uporabnikov [19]

Obvezna uporaba aktivnosti pa ima še eno posledico, ki se odraža na zadovoljstvu uporabnikov, in sicer odpor do spremembe. Odpor je naravni pojav, ki je v določeni meri prisoten pri vseh spremembah in v prvi vrsti ovira uvedbo spremembe ter povzroča nepredvidene stroške in zamude [20]. Kot že omenjeno, lahko sprejetje aktivnosti SDM vidimo kot spremembo v delovanju uporabnikov SDM. Iz psihologije je znano, da se posamezniki razlikujejo v doživljanju in odzivanju na spremembe. Nekateri spremembe sprejmejo, drugi se pasivno upirajo ali pa jih popolnoma zavračajo. Kotter in Schlesinger [22] navajata štiri glavne razloge, zakaj se ljudje upirajo spremembam:

- strah pred izgubo nečesa vrednega,

- napačno razumevanje sprememb in njenih implikacij,
- mnenje, da je sprememba nesmiselna,
- nizka tolerantnost do spremembe [21].

Posledice odpora in različnega odzivanja na spremembe pa vodijo do različnega človeškega vedenja. Smollanov model [23] opisuje tri različne odzive na spremembe. Pozitivni odzivi vodijo do zadovoljstva zaposlenih, pozitivnih čustev in prostovoljnega opravljanja pričakovanih nalog. Nevtralni odzivi so tisti, pri katerih ljudje, kljub pomislekom, privolijo v spremembe in podredljivo sodelujejo. Negativni odzivi pa vodijo do zavrnitve sprememb in nezadovoljstva uporabnikov. Vidimo lahko, da je (ne)zadovoljstvo uporabnikov neposredno povezano s prisotnostjo odpora.

Tehnična primernost in zadovoljstvo uporabnikov sta primarni meri za ocenjevanje sprejetja SDM. Za popolnost danega modela je potrebno definirati še ustrezne mere ekonomske uspešnosti uporabljane metodologije. Za osnovo lahko pri tem uporabimo empirične modele iz literature, ki preučuje vpliv informacijske tehnologije na uspešnost podjetja in literature o kriterijih uspeha projektov razvoja programske opreme. Neoklasična ekonomska teorija definira podjetje kot osnovno enoto produkcije, ki pri najmanjših možnih stroških pretvarja produkcijske faktorje (input) v dobrine in storitve (output) in pri tem maksimira dobiček [13, 24]. Spremenljivki produktivnosti (output/input) in dobičkonosnosti (profit/input) sta tako v mikroekonomskih sistemih najbolj uporabljani dimenziji uspešnosti podjetja. Med njimi sta najbolj uveljavljeni meri v empirični literaturi, ki preučuje vpliv informacijske tehnologije, dodana vrednost na zaposlenega in njena rast [25, 26]. Razlog za uveljavitev dotične mere tiči v dejstvu, da omogoča raziskovalcem, da se izognejo debatom o tem, ali podjetja res sledijo cilju maksimiranja dobička [13, 27]. Zlasti v Sloveniji obstaja precej dokazov, ki kažejo na to, da večina podjetij dobička ne maksimira [7, 12, 13], zaradi česar mere za dobičkonosnost niso primerne. V praksi pa se je tudi pokazalo [13], da managerji težko določijo dodano vrednost (output), ki jo generira izvedena

aktivnost SDM, kljub temu da je dodana vrednost glede na literaturo najboljši kriterij za ocenitev uspešnosti podjetja. Bolj oprijemljiva mera v tem primeru so stroški.

Izbira stroškov kot mere uspešnosti podjetja je v skladu z večino empiričnih modelov, ki preučujejo vpliv IT na uspešnost podjetja ter z neoklasično teorijo, a vseeno ne smemo zanemariti ostalih pomembnih organizacijskih, vedenjskih, vodstvenih in strateških teorij, ki preučujejo podjetja [13]. Pomembna razlika med njimi in neoklasično teorijo je v tem, da te teorije na podjetje gledajo kot na kompleksno združbo z več različnimi in nasprotujočimi si cilji [13, 27, 28, 29, 30, 31, 32] in ne kot na enodimenzionalno poslovno enoto, katere edini namen je večanje dobička, kot to predlaga neoklasična teorija mikroekonomije. Tako pojmovanje podjetja poleg produktivnosti omogoča merjenje še drugih dimenzij podjetniške uspešnosti.

Tak način merjenja uspešnosti izkorišča tudi široko uveljavljen sistem uravnoteženih kazalnikov (ang. *Balanced Scoreboard*), ki sta ga leta 1992 predstavila Kaplan in Norton [36]. Zanj je značilno, da merjenju tradicionalnih finančnih kazalcev doda še strateške kazalce uspešnosti, kar omogoča uravnotežen pogled na organizacijsko uspešnost podjetja. To je zagotovljeno z združevanjem štirih perspektiv:

- **Vidik učenja in rasti** se nanaša na navade podjetja pri usposabljanju in izobraževanju zaposlenih. Meritve na tem področju pomagajo usmeriti pomoč izobraževanja tja, kjer je najbolj potrebna.
- **Vidik poslovnih procesov** predstavlja izvedbo notranjih poslovnih procesov podjetja. Meritve te perspektive omogočijo managementu razumevanje, kako dobro poteka poslovanje.
- **Vidik kupcev** omogoča prepoznavanje zadovoljstva kupcev. Slabi rezultati meritev tega kazalca lahko napovedujejo poslabšanje poslovanja, kljub temu da finančni kazalci trenutni kažejo drugače.
- **Finančni vidik** je tradicionalen kazalec finančne uspešnosti, ki ima enako pomembnost tudi v sistemu uravnoteženih kazalcev.

Pregled literature nam je omogočil, da identificiramo glavne pomanjkljivosti obstoječih modelov ocenjevanja sprejetja in vpliva metodologije ter jih poskušamo vsaj delno odpraviti. Sprejetje metodologije je v literaturi največkrat obravnavano s sociološkega, psihološkega ali tehničnega vidika in se ne ozira na poslovni aspekt, ki pa je ključnega pomena pri vodstvenih odločitvah. Prav tako se obstoječe raziskave navadno osredotočajo na opazovanje metodologije kot celote, medtem ko bi bilo bolj primerno metodologijo obravnavati kot celovito enoto, ki obsega številne aktivnosti, ki so uporabljane tekom razvoja programske opreme. Tak postopek namreč omogoča prepoznavanje (ne)ustreznih delov znotraj SDM. Kot zadnje je pomembno tudi to, da se pri ocenjevanju vpliva metodologije na uspešnost podjetja ne omejimo le na stroške, temveč tudi na dodatne dimenzije podjetniške uspešnosti. Slednja dognanja bomo uporabili za gradnjo modela vrednotenja razvojnega procesa v naslednjem poglavju.

Poglavje 3

Model vrednotenja razvojnega procesa

3.1 Gradnja modela

Pregled literature nas je seznanil z obstoječimi načini ocenjevanja sprejetja in vpliva metodologije kot tudi z njihovimi pomanjkljivostmi. V tem poglavju bomo na osnovi dognanj iz prejšnjega poglavja zgradili model vrednotenja razvojnega procesa, ki bo nadgradil dosedanje modele vrednotenja SDM s tem, da naslovi njihove pomanjkljivosti.

Celovito vrednotenje SDM zahteva postavitev modela vrednotenja SDM, ki omogoča vrednotenje aktivnosti razvojnega procesa po vseh relevantnih dimenzijah. Metodologijo bomo zato obravnavali kot skupek prepletenih aktivnosti in ne kot celoto. Osredotočimo se predvsem na opazovanje takšnih aktivnosti, ki se jih da formalizirati (npr. tehnike, orodja ...) [33]. Tak način nam omogoča, da vsako aktivnost preučimo na podlagi nedvoumnih opisov. Z ocenjevanjem posameznih aktivnosti - in ne metodologije kot celote - dosežemo lažje prepoznavanje relativne primernosti posameznih aktivnosti SDM.

Aktivnosti metodologije razvoja programske opreme bomo vrednotili po socialni, tehnični in ekonomski dimenziji. Socialna dimenzija se osredotoči

na to, kako ustrezna je SDM glede na socialne značilnosti razvojne ekipe in omogoča določitev stopnje socialnega sprejetja SDM. Pod tehnično dimenzijo pa preverjamo ustreznost metodologije, glede na tehnične značilnosti projekta razvoja in/ali podjetja. Tehnična dimenzija omogoča določitev stopnje tehnične primernosti metodologije. Kot omenjamo v pregledu literature, sta Hovelja in Vavpotič v [13] stopnjo socialne sprejetosti in tehnično primernost definirala s frekvenco uporabe aktivnosti SDM in frekvenco priložnosti za uporabo aktivnosti SDM. Obe meri je bilo v našem primeru potrebno prilagoditi, saj je v opazovanem podjetju večina ključnih aktivnosti obveznih za uporabo in je posledično merjenje frekvence priložnosti nesmiselno [19]. Tehnično dimenzijo smo zato predstavili z oceno tehnične primernosti, ki jo ovrednotijo tehnične vodje. Iz istega razloga smo za merjenje socialne sprejetosti namesto FrqUse uporabili mero zadovoljstva uporabnikov SDM. Izbiro zadovoljstva uporabnikov kot primerne mere za socialno sprejetje še dodatno potrdi teorija o odporu [23], saj na ta način lahko identificiramo aktivnosti, zoper katerih je prisoten odpor uporabnikov, ki neposredno vpliva na samo socialno sprejetje metodologije.

Ekonomska dimenzija predstavlja vpliv metodologije na uspešnost podjetja. V skladu z literaturo se kot ena izmed mer uspešnosti uporabijo stroški. Druga dimenzija, ki jo navaja literatura, so cilji podjetja - v našem primeru je eden glavnih ciljev podjetja kakovosten razvoj storitve. Izbiro kakovosti razvoja storitve kot mere uspešnosti podjetja je potrdil tudi management, zato je bila ta mera vključena v zastavljeni model vrednotenja SDM. Managementu podjetja pomembno vlogo predstavlja tudi učenje in izobraževanje razvijalcev, ki je ključnega pomena pri zmanjševanju odpora do spremembe delovnih navad zaposlenih. Tretjo dimenzijo uspešnosti podjetja zato predstavlja izobraževanje in učenje razvijalcev. Nenazadnje, ker gre za razvoj informacijske storitve, je nadvse pomemben tudi odziv uporabnikov storitve. Kot zadnja izmed mer vpliva poslovne uspešnosti je bilo zato izbrano zadovoljstvo uporabnikov storitve.

Izbrane mere odstopajo od klasičnega železnega trikotnika (stroški, čas,

kvaliteta), ki se tradicionalno uporabi kot kriterij uspešnosti IT projektov [34, 35]. V našem primeru smo kriterije uspešnosti razširili, na način da vsebujejo mere organizacijsko-strateške uspešnosti in dodali še kriterije, ki jih je kot relevantne navedel management podjetja. Končne dimenzije ekonomske uspešnosti v našem modelu so:

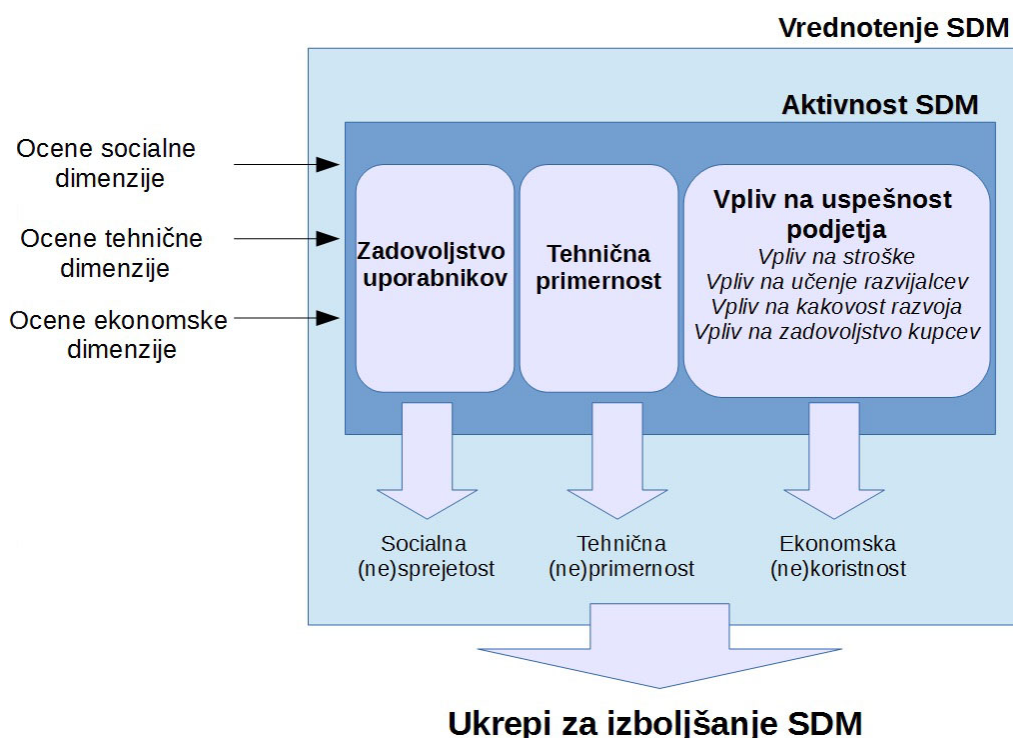
- vpliv aktivnosti na stroške podjetja,
- vpliv aktivnosti na kakovost razvoja,
- vpliv aktivnosti na učenje in izobraževanje razvijalcev,
- vpliv aktivnosti na zadovoljstvo uporabnikov.

Te štiri dimenzije predstavljajo ocenjevanje uspešnosti s sistemom uravnoteženih kazalnikov in skupaj z merami stopnje sprejetja, ki jih določata tehnična primernost in socialno sprejetje, tvorijo jedro modela za vrednotenje sprejetja in ocenitev vpliva metodologije na uspešnost podjetja.

3.2 Ocenjevanje aktivnosti

Vrednotenje razvojnega procesa se v našem primeru opravi s pomočjo treh fokusnih skupin. Prvo predstavlja management podjetja, drugo tehnične vodje in tretjo uporabniki SDM. Postopek vrednotenja razvojnega procesa je razdeljen na tri korake [13, 14].

Prvi korak zajema sistematičen popis vseh uporabljenih aktivnosti razvoja programske opreme v preučevanjem podjetju. V tem koraku fokusni skupini, ki predstavljata tehnične vodje in uporabnike SDM, sestavita seznam vseh aktivnosti SDM, ki so razvojni ekipi na voljo v sklopu določenega projekta razvoja. V procesu sodelujejo tehnične vodje - zaposleni, ki imajo celovit pogled nad tako tehničnim kot tudi poslovnim vidikom preučevanih aktivnosti in uporabniki SDM - zaposleni, ki so prisotni pri neposredni rabi omenjenih aktivnosti (razvijalci, programerji). Proces popisovanja aktivnosti je voden s strani strokovnjakov, kar zagotovi, da se res pridobi celovit in popoln seznam vseh aktivnosti.



Slika 3.1: Model vrednotenja razvojnega procesa

Drugi korak zajema ovrednotenje prej popisanih aktivnosti. V tem koraku poleg tehničnih vodij in uporabnikov SDM sodeluje tudi management podjetja. Tehnične vodje ovrednotijo aktivnosti glede na njihovo tehnično primernost, uporabniki SDM ocenijo socialno sprejetost, management podjetja pa preuči vpliv posameznih aktivnosti na uspešnost podjetja. Aktivnosti se v vseh treh primerih vrednoti na podlagi pripravljenih vprašalnikov, vidnih v Prilogi B. Vprašalniki, ki se navezujejo na tehnično primernost, od anketiranca zahtevajo, da se do posamezne trditve o tehnični primernosti opredeli z oceno na 7-točkovni Likertovi lestvici. Isti način vrednotenja uporabljajo vprašalniki, ki jih rešujejo uporabniki SDM in se navezujejo na socialno sprejetost ter vprašalniki, ki sprašujejo po vplivu aktivnosti na uspešnost podjetja. Primer trditve iz slednjega vprašalnika je npr. 'Aktivnost zelo pozitivno vpliva na nižje stroške razvoja storitve.', anketiranec pa jo ovrednoti z oceno od 1 do 7, kjer 1 predstavlja močno nestrinjanje in 7 strinjanje v celoti.

Zadnji korak je poglobljena razprava o rezultatih vrednotenja. Glede na dobljene rezultate se z managementom podjetja doreče tudi smiselne ukrepe za izboljšanje neustreznih aktivnosti procesa razvoja programske opreme.

Prednost modela je njegova fleksibilnost, saj ne obstajajo teoretični razlogi, ki bi omejevali število ovrednotenih aktivnosti SDM ali število udeležencev vrednotenja. Med razvijalci programske opreme v Sloveniji prevladujejo manjša do srednje velika podjetja in posledično je bila tudi raziskava praktičnega dela diplome opravljena v manjšem podjetju, ki se ukvarja z razvojem informacijskih storitev.

3.3 Ogrodje modela

Ocene, ki so pridobljene z vrednotenjem aktivnosti so predstavljene na razsevnem grafikonu, ki za vertikalno dimenzijo vzame mero socialne sprejetosti, za horizontalno dimenzijo pa mero tehnične primernosti. Vsaki aktivnosti v grafu je dodeljena tudi četvorka (W,X,Y,Z), ki predstavlja posamezne dimenzije vpliva aktivnosti na uspešnost podjetja. Grafikon je glede na ocenjene vrednosti razdeljen na štiri kvadrante, ki jih ločita povprečji posameznih mer. Delitev grafa na kvadrante omogoča relativno primerjavo med posameznimi aktivnostmi. Ista ocena se v različnih projektih razvoja ne preslika nujno v isti kvadrant, saj so meje kvadrantov določene dinamično, glede na povprečne vrednosti ostalih aktivnosti v projektu razvoja in/ali podjetju. Razdelitev grafa na kvadrante olajša prepoznavanje štirih tipov aktivnosti SDM [14]:

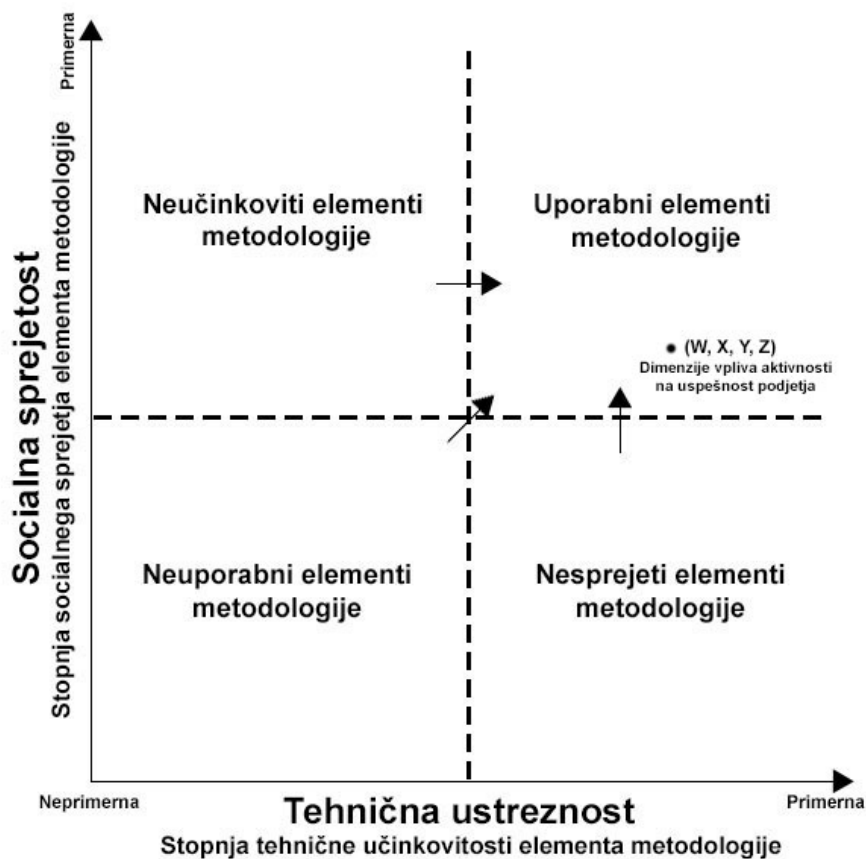
- **Neuporabna aktivnost v metodologiji** je tako tehnično kot tudi socialno neustrezna. Neustreznost je lahko posledica konstantnega napredovanja tehnologije, zaradi česar aktivnost sčasoma postane zastarela. Razvijalci jo posledično nehajo uporabljati, zaradi česar postane dokončno neustrezna. Po drugi strani je aktivnost lahko tehnično neustrezna že od vsega začetka in nikoli niti ni bil uporabljana.
- **Neučinkovita aktivnost v metodologiji** je socialno ustrezna, vendar pa ne ustreza tehničnim potrebam projekta razvoja in/ali podje-

tja. Tak primer so aktivnosti, ki so bile tehnično ustrezne na prejšnjih projektih razvoja in so jih razvijalci z veseljem uporabljali, vendar pa tehnično ne ustrezajo trenutnemu projektu razvoja, četudi bi jih razvijalci radi uporabljali.

- **Nesprejeta aktivnost v metodologiji** je obratno od neučinkovite tehnično ustrezna, vendar pa je potencialni uporabniki zaradi socialne nesprejetosti ne uporabljajo. Vzrok za to je lahko prezahtevna uporaba, nekompatibilnost s trenutnim znanjem in izkušnjami razvojne ekipe ali pa razvijalci preprosto ne vidijo prednosti njegove uporabe.
- **Uporabna aktivnost v metodologiji** je tako tehnično kot socialno ustrezna. Taka aktivnost metodologije je sprejeta s strani vseh uporabnikov in je tehnično primerna za potrebe projekta razvoja in/ali podjetja.

Za prepoznavanje vpliva aktivnosti na uspešnost podjetja so aktivnostim na grafu dodane še številske četvorke (W, X, Y, Z), ki predstavljajo vpliv aktivnosti na posamezno dimenzijo uspešnosti podjetja. Pred vsako četvorko je po formuli $(W+X+Y+Z)/4$ izračunan tudi povprečni vpliv aktivnosti na uspešnost podjetja, s katerim lažje razberemo povprečen vpliv aktivnosti na uspešnost podjetja. Glede na povprečen vpliv lahko definiramo **ekonomsko koristne** in **ekonomsko nekoristne aktivnosti**.

Zgornja kategorizacija na enostaven način identificira (ne)ustrezne aktivnosti SDM, kar pripomore k lažjemu iskanju rešitev za izboljšanje sprejetja danih aktivnosti. Potencialne rešitve definiramo kot seznam konkretnih akcij, ki naj se izvedejo, da se bo stopnja učinkovitosti ali sprejetja aktivnosti SDM izboljšala. Iskanje rešitev ne temelji zgolj na rezultatih raziskave, pač pa je za najustreznejšo rešitev potrebno dodatno znanje, ki ga pridobimo iz pogovorov z uporabniki metodologije. Diskusija je potrebna za aktivnosti, ki jih raziskava prikaže kot neustrezne. Čeprav je vsaka aktivnost različna in zahteva svojevrstne ukrepe, lahko na podlagi literature okvirno določimo splošne ukrepe, ki so razdeljeni v naslednje točke [14]:



Slika 3.2: Ogrodje modela vrednotenja SDM. Aktivnosti SDM so po vrednotenju razporejene v štiri kvadrante [14]. Po apliciranju rešitev za izboljšanje pričakujemo, da se pomemben del aktivnosti premakne v četrti kvadrant, ki predstavlja uporabne aktivnosti.

- V primeru **neučinkovite aktivnosti** moramo izboljšati tehnično primernost in hkrati ohraniti njeno socialno sprejetost in ekonomsko ustreznost. Ker so uporabniki že sprejeli uporabo aktivnosti, je potrebno poskrbeti, da aktivnost ponovno postane tehnično primerna za trenutne potrebe.
- V primeru **nesprejete aktivnosti**, ki je tehnično in ekonomsko ustrežna, je potrebno raziskati razloge za odpor uporabe aktivnosti pri upo-

rabnikih. V tem primeru razmislimo o dodatnem izobraževanju uporabnikov in predstavitvi prednosti, ki jih uporaba aktivnosti prinaša.

- V primeru **neuporabne aktivnosti**, ki je tehnično, socialno in ekonomsko neustrezna, je najbolj razumna odločitev, da aktivnost popolnoma zavržemo ali jo zamenjamo. Skoraj gotovo lahko najdemo tehnično in socialno bolj ustrezno aktivnost oz. morda taka aktivnost sploh ni potrebna
- Aktivnosti, ki se tekom vrednotenja izkažejo za **ekonomsko nekoristne** pri poskusih izboljšanja sprejetosti in tehnične primernosti ignoriramo. Take aktivnosti imajo na uspešnost podjetja zanemarljiv vpliv, zato se osredotočimo na izboljšanje sprejetosti in/ali tehnične primernosti **ekonomsko koristnih aktivnosti**.

Po apliciranju rešitev za izboljšanje pričakujemo, da se bo pomemben del aktivnosti SDM premaknil v četrti kvadrant, čeprav bodo določene aktivnosti še vedno potrebovale dodatne izboljšave ali pa kompletno zamenjavo. Dodatno je treba tudi vzeti v zakup, da je sprejetje metodologije dolgotrajen proces in so vplivi lahko opaženi šele po daljših časovnih obdobjih.

Poglavje 4

Študija primera

Raziskava je bila opravljena v manjšem slovenskem podjetju, ki se ukvarja z razvojem informacijskih storitev. Podjetje posluje že več kot 20 let, vendar na trg informacijskih storitev konkretno vstopa zadnjih 5 let. Od skupno 60 zaposlenih, podjetje zaposluje 20 razvijalcev. Storitve, ki jih ponujajo segajo od preprostih spletnih aplikacij pa do ponudbe spletne pošte in gostovanja virtualnih strežnikov. Podjetje uporablja lastno razvito metodologijo, ki predpisuje uporabo določenih procedur, aktivnosti in orodij pri izvedbi projekta razvoja. Metodologija je prilagodljiva za različne projekte, vendar pa so aktivnosti, ki jih predpisuje, največkrat obvezne za uporabo.

Zaradi prilagajanja metodologije posameznim projektom ta tudi ni formalizirana, ampak je uporabljana v obliki neformalnih dogovorov in smernic, ki jih predlaga vodstvo podjetja in/ali vodja projekta razvoja. Podjetje v namene razvoja različnih storitev ustvari razvojne ekipe, ki tipično zavzemajo 3-4 razvijalce in tehničnega vodjo, ki nadzira proces razvoja.

Koristnost zastavljenega modela vrednotenja SDM smo znotraj podjetja preverili na razvoju štirih storitev, ki imajo vsaka svojo razvojno ekipo in vodjo. Podrobnejše analize primerov sledijo v naslednjih poglavjih.

4.1 Storitev A

Storitev A predstavlja spletni vmesnik in ogrodje za enostavno kreiranje spletne strani. Ciljni kupci so uporabniki z željo po spletni strani, ki ne premorejo tehničnega znanja za konfiguracijo in vzdrževanje strežnika. Storitev trenutno uporablja 1800 uporabnikov.

Razvojno ekipo sestavljajo trije razvijalci, katerih razvoj nadzoruje tehnični vodja. Z njihovo pomočjo so bile popisane aktivnosti in rezultat je sledeč: razvoj storitve zajema 18 obveznih aktivnosti, ki narekujejo uporabo orodij, tehnik in postopkov uporabljenih med razvojem storitve. Metodologijo sestavljajo naslednje aktivnosti ¹:

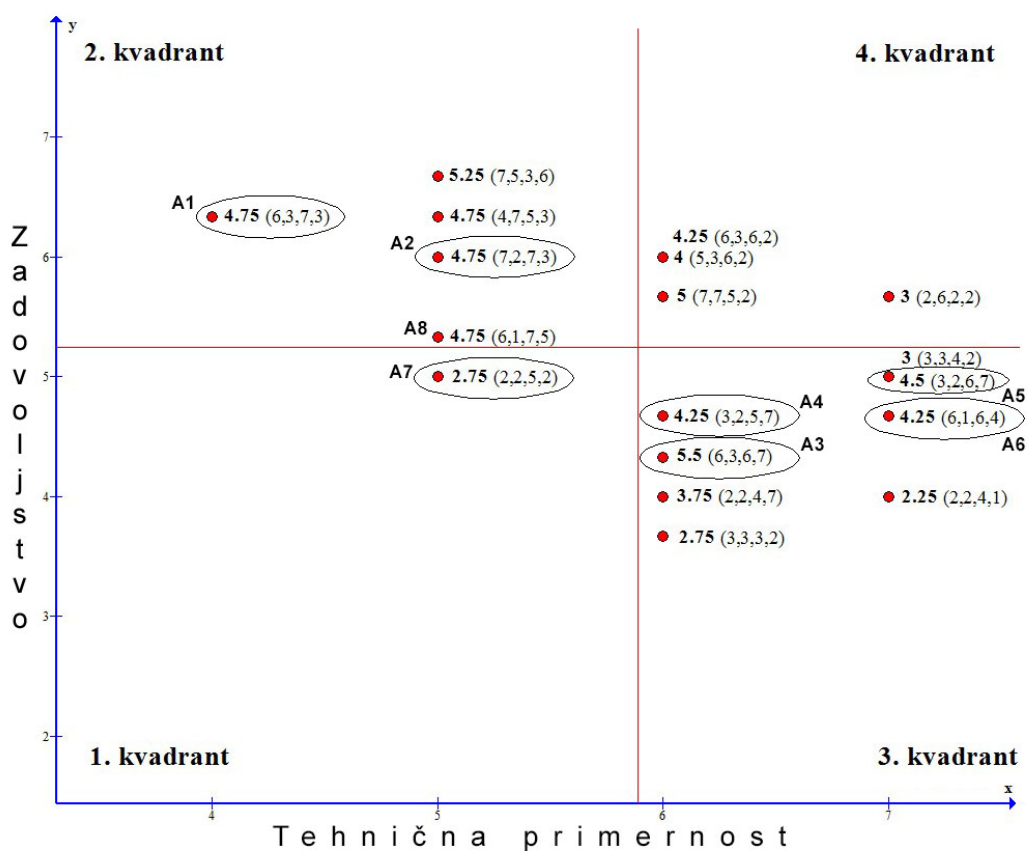
1. Sestanek projektne ekipe z naročnikom storitve
2. Interni sestanek projektne ekipe na začetku projekta
3. Izdelava načrta za izvedbo programske rešitve
4. Konfiguracija fizičnega strežnika
5. Konfiguracija spletnega strežnika z namestitvijo LAMP
6. Namestitev programske rešitve
7. Uporaba SVN za nadzorovanje različic izvirne kode
8. Dodajanje novih vtičnikov
9. Dodajanje novih tem
10. Prevajanje storitve v slovenščino
11. Ročno spreminjanje vtičnikov
12. Razvoj lastnih vtičnikov

¹Podrobnejši popis aktivnosti, ki je bil pridobljen tekom raziskave, je diplomu priložen v Prilogi A.

13. Testiranje storitve
14. Beleženje hroščev in napak
15. Pisanje dokumentacije
16. Tedenski sestanki projektne ekipe za neformalno poročanje o napredovanju projekta
17. Uporaba interne aplikacije, ki beleži zaposlenost razvijalca na določenem projektu
18. Komunikacija z uporabniki

Po samem popisu so bile aktivnosti ovrednotene na podlagi modela vrednotenja, predstavljenega v prejšnjem poglavju in s pomočjo vprašalnikov iz Priloge B. Uporabniki metodologije so ocenili svoje zadovoljstvo s posameznimi aktivnostmi, tehnične vodje so ocenili njihovo tehnično primernost, management podjetja pa je ocenil vpliv posameznih aktivnosti na uspešnost podjetja. Pri ocenjevanju vpliva aktivnosti na uspešnost podjetja se je managementu zdelo smiselno upoštevati naslednje štiri vidike: vpliv aktivnosti SDM na višjo kakovost razvoja storitve, vpliv aktivnosti SDM na nižje stroške, vpliv aktivnosti SDM na izobraževanje razvijalcev in vpliv aktivnosti SDM na zadovoljstvo uporabnikov storitve. Rezultati vrednotenja aktivnosti so predstavljeni na razsevnem grafikonu, ki ga prikazuje slika 4.1.

Podrobno preučevanje razsevnega grafa pokaže pomembne razlike med ocenami aktivnosti. Opazimo, da se nekatere aktivnosti uvrstijo v četrti kvadrant, kar jih klasificira kot uporabne aktivnosti SDM. Take aktivnosti so sprejete s strani vseh uporabnikov, prav tako pa so tehnično primerne za namene projekta razvoja. Slednje pa ne velja za aktivnosti, ki se niso uvrstile v četrti kvadrant - za njihovo izboljšanje naj bi se management osredotočil na izboljšanje bodisi socialne bodisi tehnične dimenzije aktivnosti. Predvsem pomembno je, da se naslovi ekonomsko koristne aktivnosti, ki se ne nahajajo v četrtem kvadrantu. Take aktivnosti imajo namreč največ vpliva na



Slika 4.1: Grafičen prikaz aktivnosti storitve A

samo uspešnost podjetja. Z enako pomembnostjo je potrebno identificirati ekonomsko nekoristne aktivnosti, ki se nahajajo v četrtem kvadrantu. Te so sicer socialno sprejete in tehnično primerne, vendar imajo nizek vpliv na uspešnost podjetja. Ekonomsko (ne)koristne aktivnosti smo identificirali s pomočjo managementa, ki je na podlagi vprašalnikov ovrednotil dimenzije vpliva aktivnosti na uspešnost podjetja.

Sledi predstavitev ključnih ekonomsko koristnih aktivnosti, ki se niso uvrstile v četrti kvadrant. Predstavljeni so tudi ukrepi, ki naj bi jih management izvedel, da se izboljša socialno sprejetje in tehnična primernost takih aktivnosti:

- (A1) Beleženje hroščev in napak

Aktivnost predstavlja beleženje hroščev in napak med postopkom razvoja. V podjetju za ta namen koristijo interno wikipedijo, kamor zapisujejo vse tovrstne najdbe. Kot vidimo iz grafa, so uporabniki aktivnosti z njo zelo zadovoljni, tehnično pa je aktivnost rangirana najnižje, saj wikipedija kljub svoji uporabnosti primarno ni namenjena beleženju hroščev. Aktivnost je pomembna iz vidika vpliva na uspešnost podjetja, zaradi česar je pomembno, da izboljšamo njeno tehnično primernost. Managementu se svetuje investiranje v programsko opremo za sledenje hroščem (ang. *bug-tracking*), ki uporabnikom na podoben način kot sedajšnje orodje omogoča poročanje hroščev, z naprednejšimi funkcijami pa poskrbi za njihovo enostavnejšo in hitrejšo odpravo. Poročanje hroščev bi se tako opravljalo na podoben način kot sedaj, zaradi česar bi ohranili stopnjo zadovoljstva uporabnikov, hkrati pa bi dodatne možnosti nove programske opreme zagotovile večjo tehnično primernost. Da se ne bi ocena vpliva na stroške, ki je podpovprečna, še bolj znižala, bi bilo smiselno najprej poskusiti z odprtokodnimi in brezplačnimi programskimi rešitvami.

- **(A2) Pisanje dokumentacije**

Aktivnost predstavlja dokumentiranje izvedbe procesa razvoja in pisanje dokumentacije storitve. Podobno kot pri beleženju hroščev, podjetje v ta namen koristi interno wikipedijo, ki pa je za ta namen tehnično primernejša, saj je edina naloga hranjenje pisane informacije. Razlog za slabšo tehnično oceno tiči v tem, da je dokumentacija površno spisana, hkrati pa je med razvijalci strukturno nekonsistentna. Zato je smiselno, da management pripravi jasna pravila o pisanju dokumentacije, da bo končni izdelek bolj berljiv in kvalitetnejši. Slednje bi najverjetneje imelo nekoliko negativen vpliv na zadovoljstvo uporabnikov, ki pa je trenutno ocenjeno zelo visoko, zaradi česar si lahko privoščimo izboljšanje tehnične primernosti na račun zadovoljstva uporabnikov aktivnosti. V kolikor bi tehnično primernost dvignili nad povprečni nivo,

bi se aktivnost uvrstila v četrti kvadrant.

- **(A3) Komunikacija z uporabniki**

Aktivnost predstavlja komunikacijo z uporabniki. Razvijalci telefonsko in preko e-pošte komunicirajo z uporabniki storitve. Teme pogovorov so največkrat reševanje težav pri uporabi storitve in pa tudi sprejemanje povratne informacije uporabnikov. Aktivnost ima povprečno zelo pozitiven vpliv na uspešnost podjetja, podpovprečna je edino ocena vpliva na stroške, ki predstavlja investicijo v telefonsko infrastrukturo, ki je za izvajanje aktivnost potrebna. Tehnično je aktivnost primerna, ima pa slabšo oceno zadovoljstva uporabnikov. Razvijalci namreč ne radi komunicirajo z uporabniki, saj menijo, da bi ta čas lahko bolje izkoristili. Smiselno bi bilo, da management za komunikacijo z uporabniki dodeli nov kader, kar bo razvijalcem omogočilo, da nemoteno opravljajo svoja primarna dela. Posledično bi se sicer še bolj znižala ocena stroškov, kar pa zaradi visokih ocen pri ostalih dimenzijah vpliva na uspešnost, ne bi močno vplivalo na povprečen vpliv (tudi pri najnižji oceni vpliva na stroške, bo povprečni vpliv ostal nadpovprečen glede na ostale aktivnosti).

- **(A4) Dodajanje novih vtičnikov & (A5) dodajanje novih tem**

Aktivnosti zajemata razširjanje funkcionalnosti storitve z dodajanjem novih vtičnikov in vizualnih tem. Primer take razširitve funkcionalnosti je recimo vtičnik za upravljanje z galerijami slik. Vtičniki in teme so največkrat dostopni na spletu, potrebna je zgolj njihova integracija. S slednjim razvijalci nimajo problemov, moti pa jih iskanje in testiranje primernosti novitet, ki se bodo dodale v storitev. To je tudi razlog za slabšo oceno zadovoljstva uporabnikov aktivnosti, medtem ko je tehnična primernost dobro ocenjena. Predlagana rešitev managementu je vnaprejšnja izdelava načrta (s strani tehničnega vodje), katere funkcionalnosti bo storitev potrebovala. S tem bi razvijalcem dali jasno vizijo o potrebnih funkcionalnostih in jim prihranili samostojno

iskanje primernih rešitev, kar jih trenutno tudi najbolj moti. Z dvigom zadovoljstva uporabnikov nad povprečje, bi lahko pričakovali, da se aktivnost uvrsti v četrti kvadrant.

- **(A6) Testiranje storitve**

Aktivnost predstavlja testiranje storitve pred prehodom v produkcijo, ki poteka z ročnim simuliranjem uporabniške izkušnje. Ekonomski vpliv aktivnosti je dobro ocenjen, z izjemo ocene vpliva na stroške, ki je najnižja možna in jo management zagovarja s tem, da so v času testiranja storitve pred prehodom v produkcijo vsi (tako človeški kot časovni) resursi namenjeni testiranju. Tehnično je aktivnost izvedena primerno, vendar ima podpovprečno oceno zadovoljstva uporabnikov. Določenim razvijalcem se način testiranja s simuliranjem uporabniške izkušnje ne zdi smotrni, zaradi česar je predlagana rešitev managementu, naj poskusijo z dodatno obrazložitvijo o pomembnosti tega načina testiranja. Poskusijo lahko tudi tako, da za simuliranje uporabniške izkušnje koristijo ostale zaposlene in ne razvijalce. Na ta način se lahko ugotovijo nepravilnosti, ki bi jih razvijalci zaradi njihove sorodnosti s storitvijo morda spregledali, zaradi česar bi izboljšali vpliv aktivnosti na zadovoljstvo uporabnikov storitve in kakovost razvoja.

- **(A7) Ročno spreminjanje kode vtičnikov**

Aktivnost predstavlja ročno posredovanje v kodo za spreminjanje lastnosti vtičnikov in je edina aktivnost, ki je pristala v prvem kvadrantu, kar pomeni, da je tehnično neprimerna, prav tako pa so z njeno uporabo nezadovoljni uporabniki. Ročno posredovanje v kodo vtičnikov se uporabi ob urejanju manjših programskih sprememb, vendar se razvijalci raje kot za spreminjanje obstoječih tujih vtičnikov odločajo za pisanje popolnoma lastnih. Poleg tega ima aktivnost slab vpliv na dimenzije uspešnosti podjetja - nadpovprečen doprinos ima zgolj k izobraževanju in učenju razvijalcem (ocena 5/7). Zaradi navedenih razlogov je predlagana rešitev managementu, da popolnoma ukinejo izvajanje dotične

aktivnosti, njen namen pa nadomestijo z že obstoječimi aktivnostmi, ki predstavljata razvoj lastnih vtičnikov (A8) in integracijo obstoječih vtičnikov (A4). Ti dve aktivnosti sta med uporabniki bolj priljubljeni, hkrati pa je njun vpliv na uspešnost podjetja bolje ocenjen. Iz tega razloga bi podjetje imelo večjo korist, če management nadomesti uporabo ročnega posredovanja v vtičnike z omenjenima aktivnostma.

4.2 Storitev B

Storitev B kupcem ponuja najem virtualnega strežnika za namene gostovanja dinamičnih spletnih strani. Ciljni kupci so uporabniki z željo po gostovanju spletnih strani, ki posedujejo naprednejše znanje za upravljanje s strežnikom. Storitve trenutno uporablja 1100 uporabnikov.

Razvojna ekipa je sestavljena iz štirih razvijalcev, ki jih nadzoruje tehnični vodja. Tako kot v prejšnjem primeru so razvijalci in tehnični vodja pomagali pri popisu aktivnosti. Celotna metodologija je tako sestavljena iz 22-ih (obveznih) aktivnosti, ki narekujejo uporabo orodij, tehnik in postopkov tekom razvoja storitve. Popisane so bile sledeče aktivnosti²:

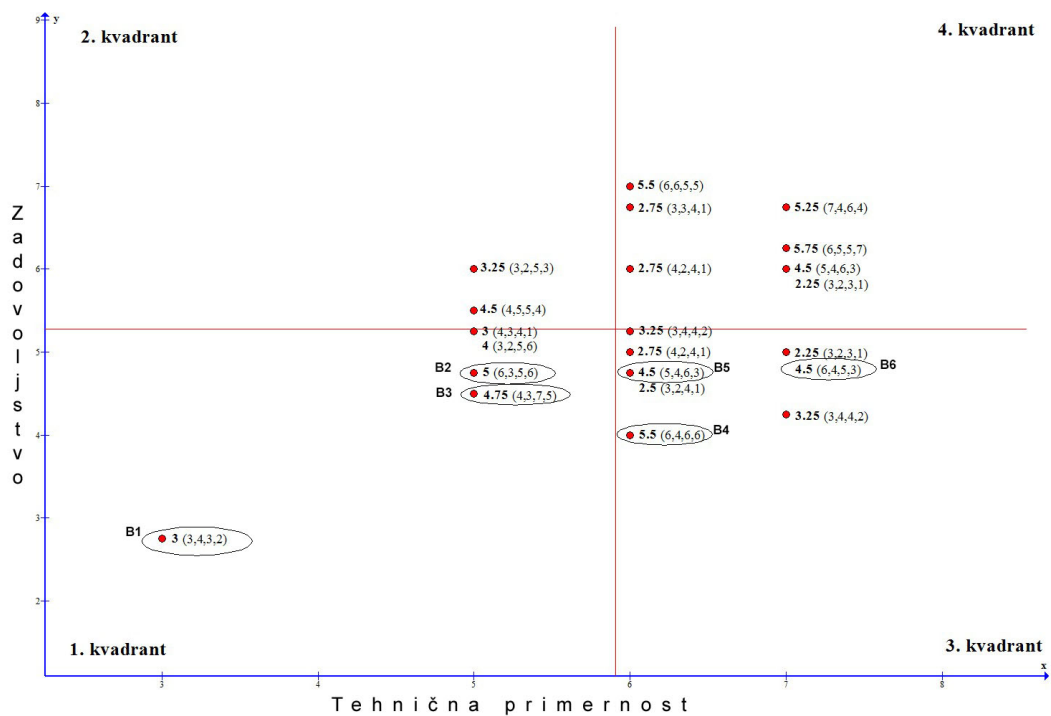
1. Sestanek projektne ekipe z naročnikom storitve
2. Interni sestanek projektne ekipe na začetku projekta
3. Načrtovanje in konfiguracija strežniške arhitekture
4. Priprava načrta za izvedbo virtualizacije
5. Virtualizacija
6. Priprava načrta za konfiguracijo uporabniških strežnikov
7. Testiranje MySQL

²Podrobnejši popis aktivnosti, ki je bil pridobljen tekom raziskave, je diplomu priložen v Prilogi A.

8. Optimizacija PHP
9. Razvoj in uporaba predloge operacijskega sistema
10. Razvoj in uporaba skripte za avtomatizacijo kreiranje strežniških instanc
11. Namestitve in uporaba strežniškega vmesnika
12. Prilagajanje strežniškega vmesnika
13. Uporaba namenske aplikacije za organizacijo izvedbe projekta
14. Razvoj in uporaba aplikacijske predloge
15. Beleženje napak in hroščev
16. Testiranje virtualnih strežnikov
17. Testiranje strežniškega vmesnika
18. Izdelava in uporaba skripte za urejevanje uporabniških profilov
19. Pisanje dokumentacije
20. Tedenski sestanki projektne ekipe
21. Uporaba interne aplikacije, ki beleži zaposlenost razvijalca na določenem projektu
22. Komunikacija z uporabniki

Po popisu aktivnosti je, tako kot pri storitvi A, sledilo ocenjevanje na podlagi vprašalnikov, ki so jih izpolnjevali uporabniki metodologije, tehnične vodje in management podjetja. Rezultati ocenjevanja so predstavljeni na razsevnem grafikonu, ki ga prikazuje slika 4.2.

Podrobno preučevanje grafa prikaže vidne razlike med ocenjenimi aktivnostmi. Opazimo, da se 7 od 22-ih aktivnosti že uvršča v četrti kvadrant.



Slika 4.2: Grafičen prikaz aktivnosti storitve B

Za razliko od prejšnjega primera pa je občutno več tudi takih aktivnosti, ki so se znašle v prvem kvadrantu (5). Za izboljšanje procesa razvoja storitve se je potrebno osredotočiti na izboljšanje socialne in tehnične dimenzije ekonomsko koristnih aktivnosti, ki se ne nahajajo v četrtem kvadrantu. Take aktivnosti imajo namreč največji vpliv na uspešnost podjetja. Z enako pomembnostjo je potrebno identificirati ekonomsko nekoristne aktivnosti, ki se nahajajo v četrtem kvadrantu, saj je njihov vpliv na uspešnost podjetja, kljub dobri sprejetosti in tehnični primernosti, nizek.

Sledi predstavitev ključnih ekonomsko koristnih aktivnosti, ki se niso uvrstile v četrti kvadrant. Predstavljeni so tudi ukrepi, ki naj bi jih management izvedel, da se izboljša socialno sprejetje in tehnična primernost takih aktivnosti:

- (B1) Uporaba interne aplikacije, ki beleži zaposlenost razvijalca na projektu & (B2) uporaba aplikacije za vodenje pro-

jektov

Aktivnost predstavlja uporabo interne aplikacije Stroškovnik, ki beleži delo razvijalca na posameznem projektu razvoja. Razvijalci morajo dnevno izpolnjevati obrazce, tako da vpisujejo, kaj so delali in koliko časa so za to porabili. Gre za zamuden proces, ki ga razvijalci vidijo kot nepotrebnega, zato slaba ocena zadovoljstva ne preseneča. Orodje je tudi precej okorno in nedodelano, zaradi česar je prejelo slabo oceno tehnične primernosti. Ekonomski vpliv aktivnosti na uspešnost podjetja je prav tako zanemarljiv. Iz teh razlogov se managementu predlaga, naj opusti izvajanje aktivnosti in jo nadomesti z uporabo programske opreme namenjene vodenju projektov. V sklopu razvoja se v ta namen že poskuša uvesti uporabo organizatorja Trello, ki kot ločena aktivnost dosega boljše ocene vpliva na uspešnost in je tehnično primernejša. Ocena zadovoljstva uporabnikov je sicer skromnejša, kar pa je posledica uporabe dveh aplikacij s podobno nalogo. Razvijalci zaradi slabe izkušnje s Stroškovnikom ne dajejo priložnosti bolj izpopolnjenim orodjem za projektno vodenje. Z ukinitvijo Stroškovnika se pričakuje povišanje zadovoljstva pri uporabi organizatorja Trello, s čimer bi aktivnost približali četrtemu kvadrantu.

- **(B3) Pisanje dokumentacije**

Aktivnost predstavlja dokumentiranje izvedbe procesa razvoja in pisanje dokumentacije storitve. Podjetje v ta namen koristi interno wiki-pedijo, ki jo zaposleni uporabljajo za odlaganje koristnih informacij. Tako zadovoljstvo uporabnikov kot tudi tehnična primernost sta podpovprečna. Ocene vpliva aktivnosti na uspešnost podjetja so dobre, zaradi česar je smiselno izboljšati stopnjo sprejetja in tehnično učinkovitost. V ta namen se managementu predlaga uporabo namenskih orodij za pisanje dokumentacije, kar bi neposredno vplivalo na tehnično primernost, posredno pa bi vpliv imelo tudi na zadovoljstvo uporabnikov. Ker je ocena vpliva na stroške nekoliko podpovprečna, se predlaga uporabo odprtokodne programske opreme. S tem bomo ohranili oceno ekonom-

ske dimenzije, hkrati pa povečali socialno in tehnično, kar bi pomenilo, da se bo aktivnost uvrstila v četrti kvadrant.

- **(B4) Komunikacija z uporabniki**

Razvijalci telefonsko in preko e-pošte komunicirajo z uporabniki storitve. Teme pogovorov so največkrat reševanje težav pri uporabi storitve in pa tudi sprejemanje povratne informacije uporabnikov. Tehnično je aktivnost primerna, vendar ne zadostuje socialnim standardom. Razvijalci namreč ne radi komunicirajo z uporabniki, saj menijo, da bi ta čas lahko boljše izkoristili. Smiselno bi bilo, da v namen povečanja zadovoljstva management za komunikacijo z uporabniki dodeli nov kader, kar bo razvijalcem omogočilo, da nemoteno opravljajo svoja primarna dela. Slednje je možno tudi iz ekonomskega vidika, saj je trenutna ocena vpliva aktivnosti na stroške zadovoljiva. Po uvedbi rešitve bi se le-ta sicer zmanjšala, vendar bi aktivnost zaradi zelo dobrih ocen ostalih dimenzij ohranila dober povprečen vpliv na uspešnost podjetja.

- **(B5) Priprava načrta za izvedbo virtualizacije**

V sklopu te aktivnosti razvojna ekipa razišče možnosti za virtualizacijo in se odloči za uporabo najprimernejše programske rešitve. Formaliziran načrt nato zapišejo na interno wikipedijo. Tehnično je aktivnost ocenjena nadpovprečno, slabše pa je zadovoljstvo uporabnikov, saj se razvijalcem formalizacija načrta ne zdi potrebna. Aktivnost pomembno prispeva h kakovosti razvoja in izobraževanju razvijalcev, zaradi česar je smiselno, da management poskusi z izboljšanjem sprejetja aktivnosti. Splošen ukrep bi v tem primeru sicer bil, da se managementu predlaga, naj programerjem podrobno predstavi pozitivne učinke formalizacije načrta, vendar ni realno pričakovati, da bodo zaradi slednjega programerji spremenili svoje mnenje. Zato je smiselno, da management poskusi z uvedbo manjših nagrad in pohval, s čimer bi programerjem dali kratkoročno motivacijo za opravljanje formalizacije načrta. Če bi bili pri tem uspešni, bi se zadovoljstvo uporabnikov povzpelo nad povprečje

in aktivnost bi se zaradi hkratne visoke tehnične primernosti uvrstila v četrti kvadrant.

- **(B6) Tedenski sestanki projektne ekipe**

Aktivnost predstavlja tedenske sestance, v sklopu katerih razvojna ekipa poroča o napredovanju razvoja storitve. S tehničnega vidika ima aktivnost najboljšo možno oceno, čemur botruje ustrezna izvedba sestankov v tehnično dovršeni sejni sobi. Prav tako je zadovoljiva povprečna ocena vpliva na uspešnost podjetja, še posebej dobro aktivnost vpliva na kakovost razvoja in izobraževanje razvijalcev. Nekoliko slabše pa je zadovoljstvo uporabnikov, saj so jim tedenski sestanki, vsaj s tako frekvenco, odveč. Menijo, da se večino stvari lahko dogovorijo tekom dela in za to ne potrebujejo sestance. Tudi v tem primeru bi se dalo na boljše zadovoljstvo uporabnikov vplivati brez drastičnih posredovanj v izvedbo aktivnosti. Zadostovala bi dodatna predstavitev pozitivnih učinkov aktivnosti. Alternativno bi lahko namesto fiksno tedenskih imeli sestance le po potrebi. S tem bi razvijalcem dali občutek, da so sestanki, kadar do njih pride, zares pomembni. Nižanje frekvence sestankovanja bi sicer imelo negativen vpliv na dimenziji kakovosti razvoja in izobraževanja razvijalcev, zaradi česar bi se nekoliko zmanjšala povprečna ocena vpliva aktivnosti na uspešnost podjetja. Zaradi sicer visokih ocen pa bi le-ta še vedno ostala nadpovprečna in povišanje zadovoljstva na njen račun je smiselno, saj bi s tem dosegli, da se aktivnost uvrsti v četrti kvadrant.

4.3 Storitev C

Storitev C je namenjena oddaljenemu shranjevanju in deljenju datotek v oblaku. Trenutno se izvaja v pilotni fazi in šteje 500 uporabnikov.

Razvojna ekipa je sestavljena iz štirih razvijalcev, ki jih nadzoruje tehnični vodja. Na enak način kot v prejšnjih primerih so bile popisane aktivnosti. Celoten proces je sestavljen iz 20-ih (obveznih) aktivnosti, ki narekujejo upo-

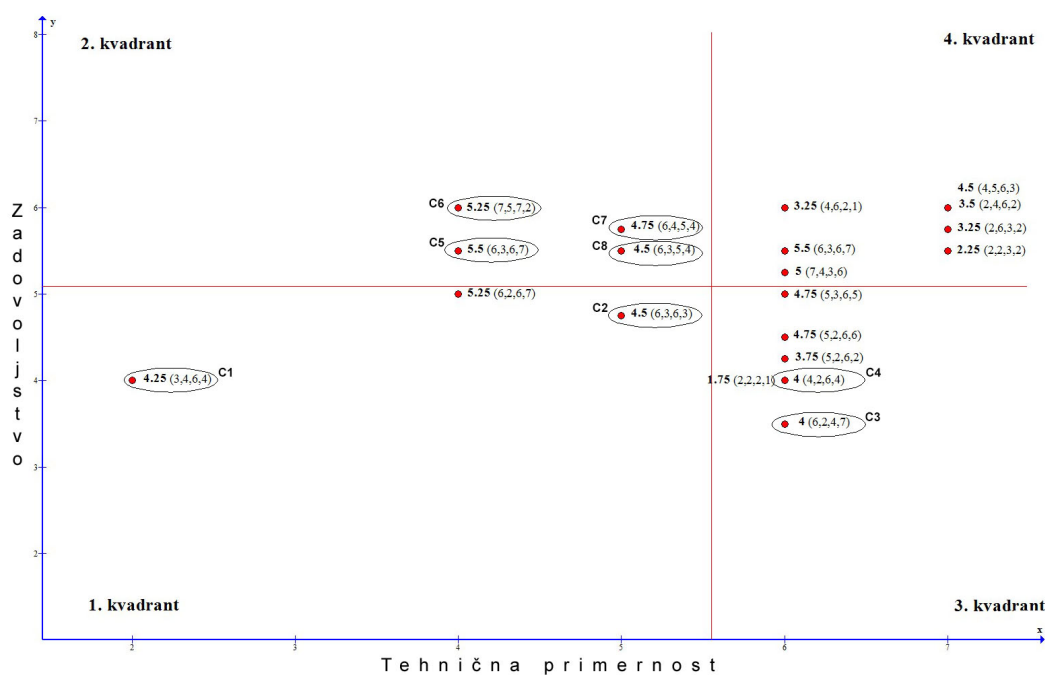
rabo orodij, tehnik in postopkov tekom razvoja storitve. Seznam popisanih aktivnosti je sledeč³:

1. Sestanek projektne ekipe z naročnikom storitve
2. Interni sestanek projektne ekipe na začetku projekta
3. Priprava načrta za realizacijo programske rešitve
4. Konfiguracija fizičnega strežnika
5. Konfiguracija spletnega strežnika z namestitvijo LAMP
6. Namestitev programske rešitve
7. Testiranje vmesnika
8. Testiranje povezljivosti z odjemalci
9. Testiranje uporabnikov s posredovanjem v bazo
10. Uporaba GIT za nadzorovanje različic izvirne kode
11. Uporaba standardnih Linux orodij
12. Prilagajanje storitve s posredovanjem v kodo
13. Prilagajanje vizualne podobe storitve
14. Prevajanje storitve
15. Pisanje dokumentacije
16. Beleženje hroščev in napak na list papirja
17. Razvoj lastnih vtičnikov

³Podrobnejši popis aktivnosti, ki je bil pridobljen tekom raziskave, je diplomu priložen v Prilogi A.

18. Tedenski sestanki projektne ekipe za neformalno poročanje o napredovanju projekta
19. Uporaba interne aplikacije Stroškovnik, ki beleži zaposlenost razvijalca na določenem projektu
20. Komunikacija z uporabniki

Popisu aktivnosti je po vzoru prejšnjih primerov sledilo ocenjevanje na podlagi vprašalnikov. Rezultati so prikazani na razsevnem grafikonu (Slika 4.3).



Slika 4.3: Grafičen prikaz aktivnosti storitve C

Graf prikaže pomembne razlike med ocenjenimi aktivnostmi. Za izboljšanje procesa razvoja storitve se je potrebno osredotočiti na izboljšanje socialne in tehnične dimenzije ekonomsko koristnih aktivnosti, ki se ne nahajajo v četrtem kvadrantu. Take aktivnosti imajo namreč največji vpliv na uspešnost

podjetja. Z enako pomembnostjo je potrebno identificirati ekonomsko nekoristne aktivnosti, ki se nahajajo v četrtem kvadrantu, saj je njihov vpliv na uspešnost podjetja, kljub dobri sprejetosti in tehnični primernosti, nizek.

Sledi predstavitev ključnih ekonomsko koristnih aktivnosti, ki se niso uvrstile v četrti kvadrant. Predstavljeni so tudi ukrepi, ki naj bi jih management izvedel, da se izboljša socialno sprejetje in tehnična primernost takih aktivnosti:

- **(C1) Beleženje hroščev in napak**

Aktivnost predstavlja beleženje hroščev in napak, ki se v tem primeru izvaja na list papirja. Ne preseneča torej, da je tehnično to najslabše ocenjena aktivnost. Boljše je sicer ocenjeno zadovoljstvo uporabnikov, saj se razvijalcem ta način dopade, ker je najhitrejši. Zatakne se v primeru razhroščevanja, ko je težko slediti postopku in popravkom. Ker aktivnost pozitivno vpliva na uspešnost podjetja, se managementu svetuje tehnična nadgradnja z uporabo namenske programske opreme za sledenje hroščem. S to rešitvijo povečamo stroške, zaradi česar bi se znižal vpliv aktivnosti na stroške. Obratno bi uporaba programske opreme za beleženje hroščev imela pozitiven vpliv na kakovost razvoja, zaradi česar ne pričakujemo, da bi ohranili najmanj enak povprečen vpliv aktivnosti na uspešnost podjetja. Pričakujemo, da bi zaradi dodatnih funkcij, ki jih ponuja programska oprema za beleženje hroščev, naraslo zadovoljstvo razvijalcev. S tem bi pozitivno vplivali tako na tehnično primernost, kot na zadovoljstvo uporabnikov, hkrati pa ohranili ekonomski vpliv aktivnosti in realno bi bilo pričakovati, da se aktivnost premakne v četrti kvadrant.

V kolikor zaradi negativnega vpliva na stroške ne bi želeli uporabljati nove programske opreme, lahko posežemo po rešitvi, ki je uporabljana na drugih projektih razvoja v podjetju, kjer v namen beleženja hroščev uporabljajo interno wikipedijo. Ta sicer ni idealna, ampak je tehnično bolj primerna od zapisovanja na list papirja. V tem primeru bi enako pozitivno vplivali na tehnično primernost, hkrati pa izboljšali oceno

vpliva na kakovost razvoja, brez da bi negativno vplivali na stroške.

- **(C2) Pisanje dokumentacije**

Aktivnost predstavlja dokumentiranje izvedbe procesa razvoja in pisanje dokumentacije. Podjetje v ta namen koristi interno wikipedijo, ki jo zaposleni uporabljajo za odlaganje koristnih informacij. Aktivnost ima velik vpliv na kakovost razvoj in izobraževanje razvijalcev, zaradi česa je smiselno izboljšati podpovprečno zadovoljstvo uporabnikov in tehnično primernost. Managementu se kot rešitev predlaga uporabo namenskih orodij za pisanje dokumentacije, kar bi neposredno vplivalo na tehnično primernost, posredno pa bi imelo vpliv tudi na zadovoljstvo uporabnikov. Dodatne funkcije, ki jih ponuja programska oprema za pisanje dokumentacije bi razvijalcem olajšale proces dokumentiranja, zaradi česar bi pričakovali povišanje v zadovoljstvu. Hkrati bi se povišal tudi vpliv aktivnosti na kakovost razvoja, zaradi česar bi se rahlo povišala tudi ocena ekonomskega vpliva aktivnosti.

- **(C3) Prevajanje storitve**

Aktivnost zajema prevajanje storitve v materni jezik. Ker bo storitev na voljo zgolj za slovenski trg, je temu primerno potrebno prilagoditi jezikovno podlago. Tehnično je aktivnost ocenjena nadpovprečno, medtem ko slabšo oceno prejme pri zadovoljstvu uporabnikov. Razvijalci se prevajanja otepajo, saj ne posedujejo primerne znanja, ki ga za opravljanje prevajanja potrebujejo. Idealna rešitev za management bi bila, da preložijo nalogo prevajanja na primeren (nov) kader, kar zaradi slabega vpliva aktivnosti na stroške ni priporočljivo. Iz vidika stroškov bi bilo zato bolj smiselno, da management dodatno poduči razvijalce. Predvsem bi pomagala razlaga prevodov uveljavljenih terminov, ki se v prevodih pogosto pojavljajo. Zaradi boljšega razumevanja procesa prevajanja lahko zato pričakujemo, da se bo zadovoljstvo uporabnikov povišalo. Hkrati bi s to rešitvijo povečali vpliv aktivnosti na izobraževanje razvijalcev, zaradi česar bi lahko zanemarili morebi-

ten negativen vpliv na stroške, ki bi ga izobraževanja imela. Zvišanje zadovoljstva uporabnikov in ohranitev ekonomskega vpliva aktivnosti, bi pomenilo, da se ob dani tehnični primernosti aktivnost uvrsti v četrti kvadrant.

- **(C4) Testiranje povezljivosti z odjemalci**

Razvijalci z ročnim simuliranjem uporabniške izkušnje testirajo povezljivost storitve z odjemalci. Tehnična primernost je zadovoljiva, vendar bi se aktivnost dalo nadgraditi z uporabo orodij za avtomatično simulacijo prometa. S tem bi tudi izboljšali zadovoljstvo uporabnikov, ki trenutno niso navdušeni na ročno simulacijo. Dodatna programska oprema bi imela negativen vpliv na stroške aktivnosti, vendar bi okrepila najmanj vpliv na kakovost razvoja. Posledično bi zaradi boljše izvedbe testiranja storitev v produkcijo šla z manj potencialnimi napakami, kar bi pomenilo, da bi aktivnost pozitivno vplivala tudi na zadovoljstvo uporabnikov storitve. S tem bi se zvišala tako ekonomska ocena kot tudi zadovoljstvo uporabnikov SDM, kar bi aktivnost uvrstilo v četrti kvadrant.

- **(C5) Prilagajanje vizualne podobe storitve**

Vizualno podobo se konfigurira z urejanjem CSS datotek. Predvidena je uporaba osnovnih (brezplačnih) orodij kot so Notepad++ ali VI. Tehnična primernost je zaradi tega nekoliko podpovprečna in z uporabo namenskih orodij za urejevanje CSS bi dotično oceno zvišali. Prav tako bi dodatne funkcije (npr. WYSIWYG) vplivale na boljšo uporabniško izkušnjo, kar bi še dodatno okrepilo zadovoljstvo uporabnikov. Uvedba nove programske opreme bi pomenila negativen vpliv aktivnosti na stroške, ki pa si ga ob trenutni oceni lahko privoščimo, saj so ostale dimenzije uspešnosti tako dobro ocenjene, da višji stroški ne bodo vplivali na nižjo ekonomsko koristnost aktivnosti.

- **(C6) Priprava načrta**

Razvojna ekipa pred začetkom dela raziše možne programske rešitve

in sestavi načrt procesa razvoja. Zapišejo ga na tablo v pisarni, tekom razvoja pa se ga posodablja in popravlja. Razvijalcem se dotična možnost dopade, saj imajo načrt vedno pred sabo, hkrati pa je urejanje preprosto in hitro. Zadovoljstvo uporabnikov s to aktivnostjo je zato prejelo visoko oceno. Slabša je tehnična primernost, saj ima zapisovanje na fizično tablo svoje pomanjkljivosti - načrta se ne da natisniti in razmnožiti, niti ni viden nikjer drugje kot v pisarni. Iz teh razlogov bi se tehnična primernost povišala že zgolj z digitalizacijo načrta. Managementu se svetuje, da se načrt v tem primeru napiše tudi na interno wikipedijo, ki jo podjetje pri drugih projektih razvoja koristi za vsehsplošno deljenje informacij. S tem bomo povišali tehnično primernost in ohranili zelo dober ekonomski vpliv aktivnosti, zaradi česar se bo aktivnost umestila v četrti kvadrant.

- **(C7) Začetni sestanek projektne ekipe & (C8) tedenski sestanki**

Aktivnosti predstavljata začetni sestanek razvojne ekipe za delitev vlog in nadaljnje tedenske sestanke za poročanje o napredovanju razvoja storitve. Zadovoljstvo uporabnikov z dotičnima aktivnostma je visoko, rahlo podpovprečna je zgolj tehnična primernost. Za izboljšanje tehnične primernosti se managementu svetuje, naj vodstvene kadre seznani z najnovejšimi moderatorskimi tehnikami, dorečejo naj tipe sestankov, konstruktivno vključujejo vse udeležence in nasploh sledijo najnovejšim trendom na področju vodenja sestankov. Izobraževanje tehničnih vodij bi imelo morebitne negativen vpliv na stroške, zaradi česar bi se znižala povprečna ocena ekonomskega vpliva aktivnosti na uspešnost podjetja. Glede na to, da aktivnosti sicer zelo pozitivno vplivata na ostale dimenzije uspešnost, bi bila ta rešitev smiselna, saj ne bi konkretno ogrozila povprečnega ekonomskega vpliva.

4.4 Storitev D

Storitev D predstavlja spletni vmesnik za pošiljanje in prejemanje elektronske pošte. Gre za najbolj razširjeno storitev podjetja, saj beleži 15000 uporabnikov.

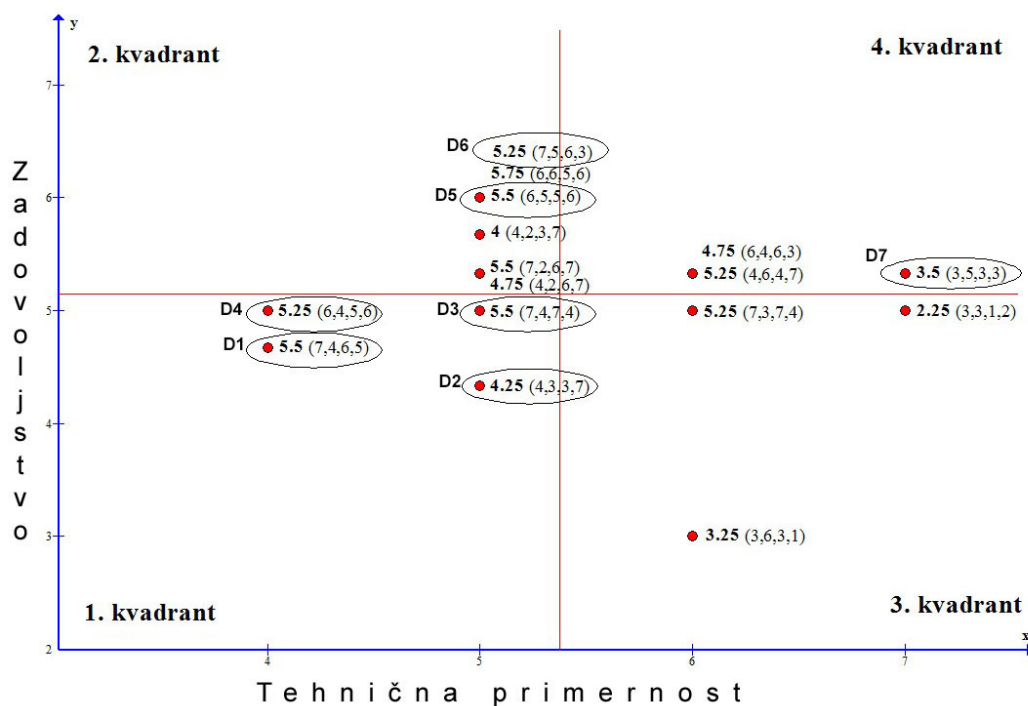
Razvojno ekipo sestavljajo tehnični vodja in trije razvijalci. Popis aktivnosti je potekal na enak način kot v prejšnjih primerih. Razvojni proces je v tem primeru sestavljen iz 16-ih obveznih aktivnosti, ki določajo uporabo orodij, tehnik in postopkov med razvojem storitve. Seznam popisanih aktivnosti je sledeč⁴:

1. Sestanek projektne ekipe z naročnikom storitve
2. Interni sestanek projektne ekipe na začetku projekta
3. Priprava načrta za izvedbo programske rešitve
4. Konfiguracija fizičnega strežnika
5. Konfiguracija spletnega strežnika z namestitvijo LAMP
6. Namestitev programske rešitve
7. Uporaba GIT za nadzorovanje različic izvirne kode
8. Razvoj lastnih vtičnikov
9. Prilagajanje vizualne podobe storitve
10. Prevajanje storitve
11. Beleženje hroščev in napak
12. Testiranje storitve
13. Pisanje dokumentacije

⁴Podrobnejši popis aktivnosti, ki je bil pridobljen tekom raziskave, je diplomu priložen v Prilogi A.

14. Tedenski sestanki projektne ekipe za neformalno poročanje o napredovanju projekta
15. Uporaba interne aplikacije Stroškovnik, ki beleži zaposlenost razvijalca na določenem projektu
16. Komunikacija z uporabniki

Popisu aktivnosti je tako kot v prejšnjih primerih sledilo ocenjevanje na podlagi vprašalnikov. Rezultati so prikazani na razsevnem grafikonu (Slika 4.4).



Slika 4.4: Grafičen prikaz aktivnosti storitve D

Graf prikaže pomembne razlike med ocenjenimi aktivnostmi. Za izboljšanje procesa razvoja storitve se je potrebno osredotočiti na izboljšanje socialne in tehnične dimenzije ekonomsko koristnih aktivnosti, ki se ne nahajajo v četrtem kvadrantu. Take aktivnosti imajo namreč največji vpliv na uspešnost

podjetja, saj je njihov vpliv na uspešnost podjetja, kljub dobri sprejetosti in tehnični primernosti, nizek.

Sledi predstavitev ključnih ekonomsko koristnih aktivnosti, ki se niso uvrstile v četrti kvadrant. Predstavljeni so tudi ukrepi, ki naj bi jih management izvedel, da se izboljša socialno sprejetje in tehnična primernost takih aktivnosti:

- **(D1) Beleženje hroščev in napak**

Kot že omenjeno pri razlagi sorodne aktivnosti pri prejšnjih storitvah, v podjetju za namen beleženja hroščev koristijo interno wikipedijo. Iz grafa je razvidna tako podpovprečna tehnična primernost kot tudi podpovprečno zadovoljstvo uporabnikov. Slednje je najverjetneje posledica slabe tehnične primernosti, saj wikipedija kot taka ni primarno namenjena beleženju hroščev in ne omogoča dodatnih funkcij, ki pomagajo pri razhroščevanju. Ker aktivnost zelo dobro vpliva na vse dimenzije uspešnosti podjetja, je potrebno poskrbeti za izboljšanje zadovoljstva uporabnikov in tehnične primernosti. Managementu je predlagana uvedba primerne programske opreme za beleženje hroščev. S tem pričakujemo zvišanje tehnične primernosti, kot posledica pa bi se dvignilo tudi zadovoljstvo uporabnikov, saj bi dodatne funkcije, ki jih prinaša primerna oprema, vplivale na lažje delo razvijalcev. Glede na solidno oceno vpliva na stroške, bi si podjetje lahko privoščilo plačljivo programsko opremo za beleženje hroščev, brez da bi konkretno vplivali na povprečen vpliv aktivnosti na uspešnost podjetja.

- **(D2) Prevajanje storitve**

Ker bo storitev na voljo zgolj za slovenski trg, je temu primerno potrebno prilagoditi jezikovno podlago. Razlog za slabo oceno pri zadovoljstvu uporabnikov je enak kot pri izvajanju podobne aktivnosti pri storitvi B: razvijalci se prevajanja otepajo, saj ne posedujejo primernega znanja, ki ga za opravljanje prevajanja potrebujejo. Pri prevajanju se zanašajo na spletna orodja kot je Google Translate, kar pripomore k slabi tehnični

primernosti. Idealna rešitev za management bi bila, da preložijo nalogo prevajanja na primeren (nov) kader, kar zaradi zmerne vpliva aktivnosti na stroške ni priporočljivo. Iz vidika stroškov bi bilo bolj smiselno, da management dodatno poduči razvijalce. Predvsem bi pomagala razlaga prevodov uveljavljenih terminov, ki se v prevodih pogosto pojavljajo. Zaradi boljšega razumevanja procesa prevajanja lahko zato pričakujemo, da se bo zadovoljstvo uporabnikov povišalo. Hkrati bi s to rešitvijo povečali vpliv aktivnosti na izobraževanje razvijalcev, zaradi česar bi lahko zanemarili morebiten negativen vpliv na stroške, ki bi ga izobraževanja imela. Zvišanje zadovoljstva uporabnikov in ohranitev ekonomskega vpliva aktivnosti, bi pomenilo, da se ob dani tehnični primernosti, aktivnost uvrsti v četrti kvadrant.

- **(D3) Pisanje dokumentacije**

Aktivnost predstavlja dokumentiranje izvedbe procesa razvoja in pisanje dokumentacije. Podjetje v ta namen koristi interno wikipedijo, ki jo zaposleni uporabljajo za odlaganje koristnih informacij. Aktivnost ima velik vpliv na kakovost razvoj in izobraževanje razvijalcev, zaradi česa je smiselno izboljšati podpovprečno zadovoljstvo uporabnikov in tehnično primernost. Managementu se kot rešitev predlaga uporabo namenskih orodij za pisanje dokumentacije, kar bi neposredno vplivalo na tehnično primernost, posredno pa bi imelo vpliv tudi na zadovoljstvo uporabnikov. Dodatne funkcije, ki jih ponuja programska oprema za pisanje dokumentacije bi razvijalcem olajšale proces dokumentiranja, zaradi česar bi pričakovali povišanje v zadovoljstvu. Hkrati bi se povišal tudi vpliv aktivnosti na kakovost razvoja, zaradi česar bi se rahlo povišala tudi ocena ekonomskega vpliva aktivnosti.

- **(D4) Sestanek projektne ekipe z naročnikom storitve & (D5) začetni sestanek projektne ekipe & (D6) tedenski sestanki**

Zadovoljstvo uporabnikov z začetnim sestankom razvojne ekipe in tedenskimi sestanki je dobro, podpovprečna pa je tehnična primernost.

V namen izboljšanja se zato managementu, podobno kot pri sorodni aktivnosti storitve C, predlaga naj vodstvene kadre izobrazi z najnovejšimi dognanji na področju vodenja sestankov. Izobraževanje tehničnih vodij bi imelo morebiten negativen vpliv na stroške, kar pa si lahko privoščimo glede na zelo dobre ocene, ki jih aktivnosti prejmejo pri drugih dimenzijah uspešnosti. Sestanek razvojne ekipe z naročnikom storitve pa je prejel podpovprečno oceno tudi pri zadovoljstvu uporabnikov. Razvijalcem se udeležba na sestanku ne zdi pretirano pomembna, saj nimajo pretirane vloge pri dialogu naročnika storitve s tehničnim vodjo. Za boljšo sprejetost aktivnosti se managementu zato svetuje, da na sestankih v dialog več vključujejo tudi razvijalce. S tem bomo povišali zadovoljstvo, in ohranili tehnično primernost ter ekonomski vpliv aktivnosti.

- **(D7) Konfiguracija spletnega strežnika z namestitvijo LAMP**

Za razliko od vseh prej omenjenih aktivnosti, je bila ta uvrščena v četrti kvadrant, saj je tehnično primerna, z njo pa so zadovoljni tudi uporabniki. Vendar pa lahko opazimo, da ekonomski vpliv aktivnosti na uspešnost podjetja nizek. V primeru ekonomsko nekoristne aktivnosti v četrtem kvadrantu je pravilno, da se vprašamo po smiselnosti njene uporabe. Konfiguracija spletnega strežnika je v procesu razvoja storitve nujno potrebna, saj bo na strežniku tekla končna programska rešitev. Izbira odprtokodnega paketa LAMP (Linux, Apache, MySQL, PHP) za konfiguracijo spletnega strežnika pripomore k dobri oceni vpliva na stroške, medtem ko na izobraževanje razvijalcev in zadovoljstvo uporabnikov storitve nima pretiranega vpliva. Z drugačno izvedbo spletnega strežnika (npr. WAMP) bi lahko vplivali na višjo kakovost razvoja, vendar bi pri tem znižali pozitiven vpliv aktivnosti na stroške. Poleg tega je zadovoljstvo uporabnikov aktivnosti trenutno visoko in bi se ob menjavi tehnologije lahko znižalo. Ker je aktivnost v procesu razvoja storitve nujna in ne obstajajo konkretne rešitve, ki bi izboljšale več dimenzij vpliva na uspešnost podjetja, se managementu

priporoča naj se aktivnost še naprej izvaja tako kot sedaj.

Poglavje 5

Sklepne ugotovitve

Diplomsko delo predstavi model za vrednotenje metodologije razvoja programske opreme. Značilnost modela je, da za ocenjevanje hkrati uporabi mere socialne sprejetosti, tehnične ustreznosti in ekonomske mere vpliva razvojnih aktivnosti na uspešnost podjetja. Tak model podjetju omogoča celovito vrednotenje celotnega procesa razvoja programske opreme iz treh vidikov (uporabniškega, tehničnega, ekonomskega). To uporabnikom modela omogoča kakovostno pripravo ukrepov za izboljšanje razvoja programske opreme. Managementu podjetja s tem omogočimo tudi boljše razumevanje in pregled na sprejetje aktivnosti.

S študijo primera je bilo dokazano, da je zgrajeni model primeren za identifikacijo (ne)primernih aktivnosti in snovanje ukrepov, ki managementu omogočajo izboljšanje razvojnega procesa. Potrdili smo primernost obravnavanja razvojnega procesa po aktivnostih in tudi to, da je smiselno za ocenjevanje metodologije uporabiti hkrati mere socialne sprejetosti, tehnične ustreznosti in mere vpliva na uspešnost podjetja.

Model je fleksibilen in jasen za uporabo, zato je naše upanje, da bo v prihodnje koristil podjetjem, ki bi si želela oceniti ali izboljšati svoje razvojne procese programske opreme.

Literatura

- [1] Avison D.E., Fitzgerald G., Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, London, 2003.
- [2] Fitzgerald B., An empirical investigation into the adoption of systems development methodologies, Information & Management 34 (1998) 317–328.
- [3] Riemenschneider C.K., Hardgrave B.C., Davis F.D., Explaining software developer acceptance of methodologies: a comparison of five theoretical models, IEEE Transactions on Software Engineering 28 (2002) 1135–1145.
- [4] Olerup A., Design approaches: a comparative study of information system design and architectural design. The Computer Journal 34, 3, (1991) 215-224.
- [5] Fitzgerald B. : Systems Development Methodologies: The Problem of Tenses. Information Technology & People, Vol. 13, No. 3, 174-185. (2000)
- [6] Belassi W. , Tukel O.I.: A new framework for determining critical success/failure factors in projects. International Journal of Project Management, Vol. 14, No. 3, 141-152. (1996)
- [7] Brezar M. : Vpliv političnega kadrovanja na poslovanje podjetij v državni lasti: analiza na primeru slovenskih družb Mercator d.d. in Petrol d.d. Ekonomska fakulteta: Ljubljana. 38. (2007)

-
- [8] Coase, R.H.: The nature of the firm. *Economica*, Vol. 4, No. 16, 386-405. (1937)
 - [9] Davis, F.A.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly* Vol. 8, No. 3, 318-339. (1989)
 - [10] Fitzgerald, B.: Systems Development Methodologies: The Problem of Tenses. *Information Technology & People*, Vol. 13, No. 3, 174-185. (2000)
 - [11] Vavpotic, D., Vasilecas, O.: An Approach for Assessment of Software Development Methodologies Suitability. *Elektron. Elektrotech.*, Vol. 8, No. 114, 107-110. (2011)
 - [12] Avison, D.E., Fitzgerald, G.: Information systems development : methodologies, techniques and tools. 4th ed. McGraw-Hill, 656. (2006)
 - [13] Vavpotic, D., Bajec, M.: An approach for concurrent evaluation of technical and social aspects of software development methodologies. *Information and Software Technology*, Vol. 51, No. 2, 528-545. (2009)
 - [14] Vavpotic, D., Bajec, M.: An approach for concurrent evaluation of technical and social aspects of software development methodologies. *Information and Software technology* 528-545 (2009)
 - [15] Rogers, E.M.: Diffusion of innovations. 5th ed. Free Press, New York, xxi, 551 p. (2003)
 - [16] Gallivan, M.J.: Organizational adoption and assimilation of complex technological innovations: development and application of a new framework. *The DATABASE for Advances in Information Systems*, Vol. 32, No. 3, 51-85. (2001)

-
- [17] Karlsson, F., Agerfalk, P.: Exploring agile values in method configuration. *European Journal of Information Systems*, Vol. 18, No. 4, 300-316. (2009)
 - [18] Lavbič, D., Lajovic, I., Krisper, M.: Facilitating information system development with Panoramic view on data. *Computer Science and Information Systems*, Vol. 7, No. 4, 737-767. (2010)
 - [19] Dwiwedi, K. Yogesh: *Information Systems Theory: Explaining and Predicting Our Digital Society*, 7-10. (2001)
 - [20] Vrhovec, S., Rupnik, R.: Obvladovanje odpora pri projektih informacijskih tehnologij. *Uporabna informatika*, XIX (2011)
 - [21] Novak, I.: *Povezanost osebnostnih lastnosti zaposlenih s pripravljenostjo na spremembo*. Ljubljana: Oddelek za psihologijo, Univerza v Ljubljani. (2012)
 - [22] Kotter, J.P. in Schlesinger, L.A. (2008). Choosing strategies for change. *Harvard business review*, 86 (7/8), 130-139.
 - [23] Smollan, K. Roy: Organizational culture, organizational change and emotions: a qualatative study. *Journal of Change Management* 9, 435-457. (2009)
 - [24] Chandler, A.D.: The Emergence of Managerial Capitalism *The Business History Review*, Vol. 58, No. 4, 473-503. (1984)
 - [25] Dedrick, J., Gurbaxani, V., Kraemer, K.L.: Information Technology and Economic Performance: A Critical Review of the Empirical Evidence. *ACM Computing Surveys (CSUR)*, Vol. 35, No. 1, 1-28. (2003)
 - [26] Dewett, T., Jones, G.R.: The role of information technology in the organization: a review, model, and assessment. *Journal of Management*, Vol. 27, No. 3, 313-346. (2001)

-
- [27] Cyert, R.M., March, J.G.: *A Behavioral Theory of the Firm* Blackwell, Oxford. (1963)
 - [28] Baumol, W.J.: *Business Behavior, Value and Growth*. MacMillan, New York, 164. (1959)
 - [29] Freeman, R.E.: *Strategic management: A stakeholder approach*. Pittman Publishing, Marshfield, MA. (1984)
 - [30] Grant, R.M.: *Toward a Knowledge-Based Theory of the Firm*. *Strategic Management Journal*, Vol. 17, No. Special Issue, 109-122. (1996)
 - [31] Perrow, C.: *The Analysis of Goals in Complex Organisations*. *American Sociological Review*, Vol. 26, No. 6, 854-866. (1961)
 - [32] Putterman, L., Kroszner, R.S.: *The Economic Nature of the Firm*. Cambridge University Press, Cambridge, 400 (1996)
 - [33] M. Fishbein, I. Ajzen, *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research*, Addison-Wesley, Reading, MA, 1975.
 - [34] Agarwal, N., Urvashi, R.: *Defining 'success' for software projects: An exploratory revelation*. *International Journal of Project Management*, Vol. 24, No. 4, 358-370. (2006)
 - [35] Atkinson, R.: *Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria*. *International Journal of Project Management*, Vol. 17, No. 6, 337-342. (1999)
 - [36] Kaplan, R. S. and Norton, D. P. (1992) *The Balanced Scorecard – Measures that Drive Performance*, *Harvard Business Review* 69 (1): 71-79.

Dodatek A

Popis aktivnosti

A.1 Storitve A

- **Sestanek razvojne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom projekta**

Pred začetkom projekta se razvojna ekipa zbere v sejni sobi, kjer jih naročnik storitve seznani z zahtevami. Skupaj zahteve preučijo, tvorijo končno idejo in zastavijo okvirne cilje projekta. Naročnik storitve je v tem primeru vodstvo podjetja, ki želi svoje storitve približati konkurenčnim. Vsa nadaljna komunikacija z naročnikom storitve je zaradi tega preprosto izvedljiva in poteka v obliki dialoga s tehničnim vodjo, po potrebi pa tudi v obliki sestankov.

- **Sestanek razvojne ekipe za razdelitev vlog in neformalno določitev načrta dela na začetku projekta**

Projektna ekipa se na začetku projekta zbere v sejni sobi, da se znotraj ekipe razdeli vloge (določi se skrbnika storitve in backup). Neformalno je dorečen tudi načrt izvedbe dela in časovni okvir razvoja. Sejna soba je tehnično dovršena, po potrebi sta na voljo računalnik in projektor. Možna je tudi video konferenca.

- **Izdelava načrta za izvedbo programske rešitve**

Razvojna ekipa razišče možne programske rešitve in se odloči za najprimernejšo (v tem primeru Wordpress). Nato sestavijo načrt o samem procesu dela in ga zapišejo na wikipedijo.

- **Konfiguracija fizičnega strežnika**

Strojna konfiguracija strežnika na katerem bo tekla storitev.

- **Konfiguracija spletnega strežnika z namestitvijo LAMP**

Na fizičnem strežniku je predvidena uporaba programskega paketa LAMP, zato razvojna ekipa poskrbi za namestitev dotične programske opreme.

- **Namestitev programske rešitve (Wordpress)**

Na strežnik se namesti izbrana programska rešitev - Wordpress.

- **Uporaba SVN za nadzorovanje različic izvirne kode**

Razvijalci napredek v razvoju shranjujejo v SVN, kar omogoča lažje sočasno delo večim ljudem na istih datotekah. S tem je poskrbljeno, da so vsi obveščeni o spremembah v programski kodi, hkrati pa je fazo razvoja v primeru problemov enostavno prestaviti na prejšnjo.

- **Dodajanje novih vtičnikov**

Wordpressu se funkcionalnost razširja z dodajanjem novih vtičnikov. Precej vtičnikov je že spisanih in objavljenih na spletu, zato jih je potrebno le integrirati v Wordpress. Razvijalci pred tem testirajo različne vtičnike in se odločijo kateri najbolj ustreza željam uporabnikov in podjetja. Možen je tudi razvoj lastnih vtičnikov, kar pa zajema samostojno aktivnost.

- **Dodajanje novih tem** Vizualno podobo v Wordpressu se uravnava preko tem. Ker uporabniki storitve želijo čim bolj raznolike spletne strani, razvijalci v storitev integrirajo uporabo različnih tem. Le-te so že napisane in javno dostopne na spletu, potrebna je zgolj integracija v storitev.

- **Prevajanje storitve v slovenščino**

Jedro Wordpressa je načeloma že prevedeno, potrebni so zgolj manjši popravki v prevodih. Več dela je z dodanimi temami, ki niso lokalizirane, zaradi česar jih je za intuitivno uporabo potrebno prevesti. Razvijalci se zanašajo na lastno znanje angleščine in uporabo spletnih prevajalskih orodij (Google Translate, Računalniški slovarček). Prevaja se preko vmesnika Codestyling Localisation.

- **Konfiguracija vtičnikov**

Obstoječe vtičnike razvijalci prilagodijo potrebam podjetja z ročnim posredovanjem v kodo. Predvidena je uporaba Eclipsea, programski jezik pa je PHP.

- **Razvoj lastnih vtičnikov**

V kolikor se izkaže, da željen vtičnik še ne obstaja, ali pa je obstoječ neprimeren za uporabo, razvijalci napišejo lasten vtičnik. Predvidena je uporaba Eclipsea, programski jezik pa je PHP.

- **Testiranje storitve**

Testiranje storitve poteka z ročnim simuliranjem uporabniške izkušnje. V kolikor so odkrita nedelovanja, se jih zapiše na wikipedijo.

- **Beleženje hroščev in napak na wikipedijo**

Razvijalci tekom razvoja storitve vse napake, ki jih odkrijejo zapisujejo na wikipedijo. Če najdejo rešitev za težavo, jo dopišejo (kar pa se nemalokrat pozabi).

- **Dokumentiranje izvedbe procesa**

Dokumentacija postopka razvoja storitve se piše na interno wikipedijo, ki je dostopna vsem zaposlenim v podjetju. Hkrati se objavljajo tudi navodila za uporabo storitve in z njo povezane informacije. S tem se zagotovi, da so ob ponovni uporabi funkcij iz razvoja storitve informacije o le-teh celovito in hitro dostopne.

- **Tedenski sestanki razvojne ekipe za neformalno poročanje o napredovanju projekta**

Razvojna ekipa se vsak teden zbere v sejni sobi za poročanje o napredovanju projekta. Pisna poročila niso potrebna, dovolj je govorna ob-razložitev tehničnemu vodji o stanju projekta. Sejna soba je tehnično dovršena, po potrebi sta na voljo računalnik in projektor. Možna je tudi videokonferenca.

- **Uporaba interne aplikacije Stroškovnik, ki beleži zaposlenost razvijalca na določenem projektu**

Aplikacija Stroškovnik vodi evidenco dela razvijalca na različnih projekti. Razvijalci morajo za vsak delovni dan izpolniti koliko časa, na katerem projektu in kaj konkretno so delali. Te informacije se uporablja za statistiko ter nadzor dela na globalnem nivoju podjetja.

- **Komunikacija z uporabniki**

Kot sredstvo za prenos informacije med uporabniki storitev in razvijalci se uporabljata telefon in elektronska pošta. Teme pogovorov so reševanje težav pri uporabi storitve in odpravljanje napak. Prav tako se tovrstno komunikacijo izkorišča za sprejemanje

A.2 Storitev B

- **Sestanek razvojne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom projekta**

Glej Priloga A - Storitve A.

- **Sestanek razvojne ekipe za razdelitev vlog in neformalno določitev načrta dela na začetku projekta**

Glej Priloga A - Storitve A.

- **Načrtovanje in konfiguracija strežniške arhitekture z uporabo orodja za avtomatizacijo Puppet**

Na fizični strežnik se namesti orodje Puppet, ki je namenjeno avtomatizaciji ureditve strežniškega sistema. Na strežniku se nato kreirajo t. i. recepti, preko katerih se nadzira stanje vseh podrejenih enot v infrastrukturi. Na ta način se repetativne naloge izvedejo samo enkrat, Puppet pa poskrbi, da se spremembe uveljavijo na vseh (virtualnih) strežnikih.

- **Priprava načrta za izvedbo virtualizacije**

Razvojna ekipa razišče možne rešitve za virtualizacijo in se odloči za najprimernejšo (Virtuozzo). Nato okvirno splanirajo potek virtualizacije in načrt zapišejo na wikipedijo.

- **Virtualizacija z uporabo Virtuozza**

Za virtualizacijsko platformo raziskovalci namestijo in uporabljajo Virtuozzo.

- **Priprava načrta za konfiguracijo uporabniških strežnikov**

Razvijalci preučijo razpoložljive tehnologije in pripravijo načrt o programski izvedbi uporabniških strežnikov. Na tej točki je okvirno dorečena izbira odprtokodne rešitve LAMP (Linux Centos, Apache, MySQL, PHP) in strežniškega vmesnika Ajenti. Za natančno izbiro verzij MySQL in PHP so potrebna dodatna testiranja, ki tvorijo samostojne aktivnosti. Načrt se formalizira in zapiše na interno wikipedijo.

- **Testiranje MySQL in podatkovne baze**

Razvijalci se s testiranjem posameznih verzij MySQL odločijo katera bi bila najprimernejša za uporabo na strežnikih. Testiranje poteka z ročnim posredovanjem v bazo, ali pa z uporabo JMeter, ki simulira uporabnike in njihove zahteve.

- **Optimiziranje PHP**

Razvijalci se s testiranjem pospeševalcev optimizirajo PHP na hitrejšo delovanje. Testiranje poteka tako, da se uporabi različne pospeševalce

in nato ročno beleži odzivne čase. Pospeševalec, ki se najbolj odreže na testih, se nato uporabi pri nadaljnjem razvoju.

- **Razvoj in uporaba predloge operacijskega sistema, ki se uporablja za avtomatično konfiguracijo uporabniških strežnikov**

Razvoj bash skripte, ki kot vhodno datoteko vzame seznam s programsko opremo in repozitoriji, kjer se nahaja. Skripta nato programsko opremo prenese iz repozitorijev in naredi sliko (image), ki se uporabi kot predloga za namestitev LAMP na svež strežnik. Z uporabo predloge je poskrbljeno, da so na vseh strežnikih uporabljane enake tehnologije, izogne pa se tudi ročnem nameščanju le-teh. Za pisanje skripte se uporabi osnovno Linux orodje VIM editor.

- **Razvoj skripte za avtomatizacijo kreiranja strežniških instanc**

Aktivnost zajema proces izdelave skripte, ki avtomatizira kreiranje strežniške instance, kar pripomore k hitrejši izvedbi procesa, saj je instance sicer potrebno kreirati ročno, kar pa je napram avtomatizaciji zamuden proces. Skripta je napisana v skriptnem jeziku bash, za pisanje pa se uporabi Linux orodje VIM.

- **Namestitev strežniškega vmesnika Ajenti**

Vmesnik Ajenti skrbi za lažje upravljanje z uporabniškimi strežniki. Potrebna je namestitev, ki se jo izvede ročno z uporabo zaporedja Linux ukazov in s popravljanjem namestitvenih datotek v tekstovnem urejevalniku.

- **Prilagajanje strežniškega vmesnika Ajenti**

Strežniški vmesnik je potrebno prilagoditi za potrebe podjetja; menjava barvne sheme, prevod v materni jezik, in dodajanje lastnih funkcij. Vse popravke se izvaja s pisanjem bash skript in ročnim popravljanjem programske kode v že obstoječih datotekah, za reševanje pogostih problemov pa se razvijalci zatečejo tudi na GitHub.

- **Uporaba organizatorja Trello za organizacijo izvedbe projekta**

Trello je brezplačna aplikacija za organiziranje projektov. Znotraj projekta je uporabljana za razdelitev projekta na naloge in podnaloge, katerim se določi odgovorne osebe. Aktivnost skrbi za organizacijo in evidenco opravljenih nalog v projektu.

- **Razvoj in uporaba aplikacijske predloge za avtomatizirano nameščanje programske opreme na uporabniške strežnike**

V ločenih tekstovnih dokumentih se zbere seznam željene programske opreme in seznam repozitorijev, kjer se jo dobi. Nato se napiše bash skripto, ki te dokumente prebere, vsebino prenese iz repozitorijev in iz celotnega paketa ustvari sliko (image), ki se jo nato namesti na posamezno instanco strežnika. Predloga predstavlja avtomatiziran proces namestitve programske opreme.

- **Beleženje napak in hroščev na wikipedijo**

Vse težave, napake in hrošče, ki se jih odkrije med razvojem storitve je potrebno skupaj z rešitvijo zapisati na interno wikipedijo. S tem se doseže, da so vsi razvijalci seznanjeni s problemi in njihovo rešitvijo, kar omogoča hitrejšo odpravo napak pri morebitni ponovitvi problema.

- **Testiranje virtualnih strežnikov**

Ročno se testira povezovanje na strežnike, uporabo odjemalcev in nameščanje aplikacij. Velik poudarek je tudi na prebiranju dnevniških datotek za iskanje morebitnih napak.

- **Testiranje strežniškega vmesnika Ajenti**

Aktivnost zajema ročno pregledovanje programske kode strežniškega vmesnika kot tudi testiranje vmesnika samega s simuliranjem uporabniških zahtev.

- **Razvoj in uporaba skripte za urejevanje uporabniških profilov**

V izogib ročnemu popravljanju podatkov v uporabniških profilih se napiše skripto, ki omogoča enostavnejše urejanje in tako prihrani čas, ki

bi bil potreben za ročno ponastavljanje atributov. Skripta je napisana v bashu, z uporabo VIM urejevalca.

- **Pisanje dokumentacije in navodil storitve**

Dokumentacija postopka razvoja storitve se piše na interno wikipedijo, ki je dostopna vsem zaposlenim v podjetju. Hkrati se objavljajo tudi navodila za uporabo storitve oziroma z njo povezanih uporabnih postopkov. S tem se zagotovi, da so ob morebitni ponovni uporabi funkcij/postopkov iz razvoja storitve informacije o le-teh celovito in hitro dostopne.

- **Tedenski sestanki razvojne ekipe za neformalno poročanje o napredovanju projekta**

Glej Priloga A - Storitve A.

- **Uporaba interne aplikacije Stroškovnik, ki beleži zaposlenost razvijalca na določenem projektu**

Glej Priloga A - Storitve A.

- **Komunikacija z uporabniki**

Glej Priloga A - Storitve A.

A.3 Storitve C

- **Sestanek razvojne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom projekta**

Glej Priloga A - Storitve A.

- **Sestanek razvojne ekipe za razdelitev vlog in neformalno določitev načrta dela na začetku projekta**

Glej Priloga A - Storitve A.

- **Priprava načrta za realizacijo programske rešitve**

Razvojna ekipa raziše možne programske rešitve in se odloči za najprimernejšo (v tem primeru Owncloud). Nato sestavijo načrt procesa dela in ga zapišejo na tablo v pisarni. Tekom procesa razvoja pa se ga posodablja in popravlja.

- **Konfiguracija fizičnega strežnika**

Glej Priloga A - Storitve A.

- **Konfiguracija spletnega strežnika z namestitvijo LAMP**

Glej Priloga A - Storitve A.

- **Namestitev programske rešitve (Owncloud)**

Na strežnik se namesti izbrana programska rešitev - Owncloud.

- **Testiranje vmesnika**

Testiranje vmesnika poteka ročno, s klikanjem po grafični podobi razvijane storitve. Morebitne napake in hrošče si razvijalci zapisujejo na list papirja.

- **Testiranje povezljivosti z odjemalci**

Testira se obnašanje storitve pri povezovanju z odjemalcem. Razvijalci ročno z uporabo različnih odjemalcev simulirajo uporabniško izkušnjo. Alternativa so orodja za simulacijo prometa, ki pa se ne uporabljajo pogosto.

- **Testiranje uporabnikov**

Razvijalci testirajo uporabniške profile in delajo popravke z ročnim posredovanjem v bazo.

- **Uporaba GIT za nadzorovanje različic izvirne kode**

Vsi razvijalci med programiranjem uporabljajo GIT, ki omogoča neovirano sočasno delo večim ljudem na istih datotekah. S tem je poskrbljeno, da so vsi obveščeni o spremembah v programski kodi, hkrati pa je fazo razvoja v primeru problemov enostavno prestaviti na prejšnjo.

- **Uporaba standardnih Linux orodij**

Razvijalci pri svojem delu ne uproabljaajo namenskih programov, ampak se večinoma zanašajo na standardna Linux orodja (ssh, vim, tcpdump ...), ki v večini zadostujejo potrebam tekom razvoja.

- **Prilagajanje Ownclouda s posredovanjem v kodo**

Razvijalci osnovno rešitev Owncloud prilagodijo potrebam podjetja. Sem spada ročno popravljanje kode in pisanje lastnih funkcij v PHPju. Predvidena je uporaba Eclipsea.

- **Prilagajanje vizualne podobe Ownclouda**

Poskrbeti je treba, da vizualna podoba storitve ustreza profilu podjetja. Razvijalci slednje dosežejo tako, da napišejo lastne CSS datoteke in jih integrirajo v Owncloud.

- **Prevajanje Ownclouda**

Zaradi trženja storitve na slovenski trg je storitev treba prevesti v materni jezik. Razvijalci si pomagajo z lastnim znanjem angleščine in uporabo spletnih orodij za prevajanje (Google Translate, Računalniški slovarček).

- **Dokumentiranje izvedbe procesa**

Dokumentacija postopka razvoja storitve se piše na interno wikipedijo, ki je dostopna vsem zaposlenim v podjetju. Hkrati se objavljajo tudi navodila za uporabo storitve oziroma z njo povezane koristne informacije. S tem se zagotovi, da so ob morebitni ponovni uporabi funkcij iz razvoja storitve informacije o le-teh celovito in hitro dostopne.

- **Beleženje hroščev in napak na list papirja**

Razvijalci tekom razvoja storitve vse napake, ki jih odkrijejo zapisujejo na list papirja, saj je to najhitrejše. V skrajnih primerih se hrošče zapiše tudi v tekstovni urejevalnik Notepad.

- **Razvoj lastnih vtičnikov**

Owncloudu se razširja funkcionalnosti z dodajanjem vtičnikov. Razvijalci zato po potrebi napišejo funkcionalne programe (vtičnike) in jih dodajo storitvi. Predvidena je uporaba Eclipsea, programski jezik pa je PHP. Tak primer je recimo vtičnik za poenoteno prijavo.

- **Tedenski sestanki razvojne ekipe za neformalno poročanje o napredovanju projekta**

Glej Priloga A - Storitve A.

- **Uporaba interne aplikacije Stroškovnik, ki beleži zaposlenost razvijalca na določenem projektu**

Glej Priloga A - Storitve A.

- **Komunikacija z uporabniki**

Glej Priloga A - Storitve A.

A.4 Storitev D

- **Sestanek razvojne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom**

projekta

Glej Priloga A - Storitve A.

- **Sestanek razvojne ekipe za razdelitev vlog in neformalno določitev načrta dela na začetku projekta**

Glej Priloga A - Storitve A.

- **Priprava načrta za izvedbo programske rešitve**

Razvojna ekipa razišče možne programske rešitve in se odloči za najprimernejšo (v tem primeru Roundcube). Nato sestavijo načrt o samem procesu dela in ga zapišejo na wikipedijo.

- **Konfiguracija fizičnega strežnika**

Glej Priloga A - Storitve A.

- **Konfiguracija spletnega strežnika z namestitvijo LAMP**

Glej Priloga A - Storitve A.

- **Namestitev programske rešitve (Roundcube)**

Na strežnik se namesti izbrana programska rešitev - Roundcube.

- **Uporaba GIT za nadzorovanje različic izvirne kode**

Vsi razvijalci med programiranjem uporabljajo GIT, ki omogoča sočasno delo večim ljudem na istih datotekah. S tem je poskrbljeno, da so vsi obveščeni o spremembah in programski kodi, hkrati pa je fazo razvoja v primeru problemov enostavno prestaviti prejšnjo.

- **Razvoj lastnih vtičnikov**

Roundcubu se razširja funkcionalnosti z dodajanjem vtičnikov. Razvijalci zato po potrebi napišejo funkcionalne programe (vtičnike) in jih dodajo storitvi. Predvidena je uporaba Dreamweaverja in VI-ja, programski jezik pa je PHP. Tak primer je recimo vtičnik za avtomatski odgovor.

- **Prilagajanje vizualne podobe Roundcuba**

Razvijalci videz storitve prilagodijo z urejanjem CSS in HTML datotek. Predvidena je uporaba Dreamweaverja in VI-ja. Za testiranje in urejanje videza v realnem času si nekoliko pomagajo tudi z uporabo Firebuga in Webinspectorja.

- **Prevajanje Roundcuba**

Zaradi trženja storitve na slovenski trg je storitev treba prevesti v materni jezik. Razvijalci si pomagajo z lastnim znanjem angleščine in uporabo spletnih prevajalnikov (Google Translate, Računalniški slovarček).

- **Beleženje hroščev in napak na wikipedijo**

Razvijalci tekom razvoja storitve vse napake, ki jih odkrijejo zapisujejo na wikipedijo, hkrati pa jih objavijo tudi na interno mailing listo, kjer skupaj napake tudi rešijo.

- **Testiranje storitve**

Testiranje poteka z ročnim simuliranjem uporabniške izkušnje. V kolikor so odkrita nedelovanja, se jih zapiše na wikipedijo.

- **Tedenski sestanki razvojne ekipe za neformalno poročanje o napredovanju projekta**

Glej Priloga A - Storitve A.

- **Uporaba interne aplikacije Stroškovnik, ki beleži zaposlenost razvijalca na določenem projektu**

Glej Priloga A - Storitve A.

- **Komunikacija z uporabniki**

Glej Priloga A - Storitve A.

Dodatek B

Primer vprašalnika za management

Namen vprašalnika je ovrednotenje vpliva posamezne aktivnosti znotraj metodologije na uspešnost podjetja. Dimenzije uspešnosti zajemajo vpliv na kakovost razvoja, vpliv na stroške, vpliv na učenje in izobraževanje razvijalcev ter vpliv na zadovoljstvo uporabnikov.

Anketiranec se do vsake trditve opredeli z oceno na lestvici 1-7, kjer velja:

- 1 - sploh se ne strinjam,
- 2 - ne strinjam se,
- 3 - večinoma se ne strinjam,
- 4 - niti se ne strinjam niti se strinjam,
- 5 - večinoma se strinjam,
- 6 - strinjam se,
- 7 - v celoti se strinjam.

- **Sestanek projektne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom projekta**

Sestanek projektne ekipe z naročnikom storitve zelo pozitivno vpliva na višjo kakovost razvoja storitve.

1 2 3 4 5 6 7

Sestanek projektne ekipe z naročnikom storitve zelo pozitivno vpliva na nižje stroške razvoja storitve.

1 2 3 4 5 6 7

Sestanek projektne ekipe z naročnikom storitve zelo pozitivno vpliva na izobraževanje in učenje razvijalcev.

1 2 3 4 5 6 7

Sestanek projektne ekipe z naročnikom storitve zelo pozitivno vpliva na zadovoljstvo uporabnikov storitve.

1 2 3 4 5 6 7

- **Priprava načrta za izvedbo programske rešitve**

Izdelava načrta za izvedbo programske rešitve zelo pozitivno vpliva na višjo kakovost razvoja storitve.

1 2 3 4 5 6 7

Izdelava načrta za izvedbo programske rešitve zelo pozitivno vpliva na nižje stroške razvoja storitve.

1 2 3 4 5 6 7

Izdelava načrta za izvedbo programske rešitve zelo pozitivno vpliva na izobraževanje in učenje razvijalcev.

1 2 3 4 5 6 7

Izdelava načrta za izvedbo programske rešitve zelo pozitivno vpliva na zadovoljstvo uporabnikov storitve.

1 2 3 4 5 6 7

- **Beleženje hroščev in napak na wikipedijo**

Beleženje hroščev in napak zelo pozitivno vpliva na višjo kakovost razvoja storitve.

1 2 3 4 5 6 7

Beleženje hroščev in napak zelo pozitivno vpliva na nižje stroške razvoja storitve.

1 2 3 4 5 6 7

Beleženje hroščev in napak zelo pozitivno vpliva na izobraževanje in učenje razvijalcev.

1 2 3 4 5 6 7

Beleženje hroščev in napak zelo pozitivno vpliva na zadovoljstvo uporabnikov storitve.

1 2 3 4 5 6 7

- **Testiranje storitve**

Testiranje storitve zelo pozitivno vpliva na višjo kakovost razvoja storitve.

1 2 3 4 5 6 7

Testiranje storitve zelo pozitivno vpliva na nižje stroške razvoja storitve.

1 2 3 4 5 6 7

Testiranje storitve zelo pozitivno vpliva na izobraževanje in učenje razvijalcev.

1 2 3 4 5 6 7

Testiranje storitve zelo pozitivno vpliva na zadovoljstvo uporabnikov storitve.

1 2 3 4 5 6 7

Dodatek C

Primer vprašalnika za tehnične vodje

Namen vprašalnika je ovrednotenje tehnične dovršenosti posamezne aktivnosti znotraj metodologije (SDM - software development methodology).

Anketiranec oceni kako tehnično dovršena je izvedba ali uporaba posamezne aktivnosti SDM, tako da se do vsake trditve opredeli z oceno na lestvici 1-7, kjer velja:

- 1 - sploh se ne strinjam,
- 2 - ne strinjam se,
- 3 - večinoma se ne strinjam,
- 4 - niti se ne strinjam niti se strinjam,
- 5 - večinoma se strinjam,
- 6 - strinjam se,
- 7 - v celoti se strinjam.

- **Sestanek projektne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom projekta.**

Sestanek projektne ekipe z naročnikom storitve poteka v skladu z naj-novejšimi smernicami in dognanji stroke na tem področju.

1 2 3 4 5 6 7

- **Priprava načrta za izvedbo programske rešitve**

Izdelava načrta za izvedbo programske rešitve poteka v skladu z najnovejšimi smernicami in dognanji stroke na tem področju.

1 2 3 4 5 6 7

- **Beleženje hroščev in napak na wikipedijo**

Beleženje hroščev in napak poteka v skladu z najnovejšimi smernicami in dognanji stroke na tem področju.

1 2 3 4 5 6 7

- **Testiranje storitve**

Testiranje storitve poteka v skladu z najnovejšimi smernicami in dognanji stroke na tem področju.

1 2 3 4 5 6 7

Dodatek D

Primer vprašalnika za uporabnike SDM

Namen vprašalnika je ovrednotenje socialne sprejetosti posamezne aktivnosti znotraj metodologije (SDM - *software developement methodology*). Dimenzijo socialne sprejetosti predstavlja *zadovoljstvo uporabnikov aktivnosti SDM*.

Anketiranec oceni vpliv posamezne aktivnosti SDM na zadovoljstvo uporabnikov metodologije, tako da se do vsake trditve opredeli z oceno na lestvici 1-7, kjer velja:

- 1 - sploh se ne strinjam,
- 2 - ne strinjam se,
- 3 - večinoma se ne strinjam,
- 4 - niti se ne strinjam niti se strinjam,
- 5 - večinoma se strinjam,
- 6 - strinjam se,
- 7 - v celoti se strinjam.

- **Sestanek projektne ekipe z naročnikom storitve za preučevanje zahtev, določitev ciljev in diskutiranje o ideji pred začetkom projekta.**

Sestanek projektne ekipe z naročnikom zelo pozitivno vpliva na zadovolj-

stvo članov projektne ekipe, saj pomembno prispeva k učinkovitosti dela, informiranosti o delovnih nalogah ter k dobremu delovnemu vzdušju.

1 2 3 4 5 6 7

- **Priprava načrta za izvedbo programske rešitve**

Izdelava načrta za izvedbo programske rešitve zelo pozitivno vpliva na zadovoljstvo članov projektne ekipe, saj pomembno prispeva k učinkovitosti dela, informiranosti o delovnih nalogah ter kvalitetnejšemu razvoju storitve.

1 2 3 4 5 6 7

- **Beleženje hroščev in napak na wikipedijo**

Beleženje hroščev in napak zelo pozitivno vpliva na zadovoljstvo članov projektne ekipe, saj je ustrezno izvedeno in pomembno prispeva k učinkovitosti dela in kvalitetnejšemu razvoju storitve.

1 2 3 4 5 6 7

- **Testiranje storitve**

Testiranje storitve zelo pozitivno vpliva na zadovoljstvo članov projektne ekipe, saj je ustrezno izvedeno in pomembno prispeva k učinkovitosti dela in kvalitetnejšemu razvoju storitve.

1 2 3 4 5 6 7